University of Alberta

WHAT TO DO WHEN YOU DON'T HAVE MUCH DATA: ISSUES IN SMALL SAMPLE PARAMETER LEARNING IN BAYESIAN NETWORKS

by

Ajit Paul Singh

©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Spring 2004

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-96547-3
Our file Notre référence
ISBN: 0-612-96547-3

The author has granted a non-exclusive license allowing the Library and Archives Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

A discrete Bayesian network is a factorization of a joint distribution over random variables. The most common use of these networks is for the computation of conditional probabilities (query responses). The parameters of these networks are often learned from data. Thus network parameters are themselves uncertain, which induces a distribution for any query response. When data sets are small, the effects of parameter uncertainty can be severe. In this thesis we argue that when data sets are small, the distribution of a query response is accurately modelled by a Beta distribution. Procedures for the modeling of the query response are also reviewed. Furthermore, we examine proposed techniques for parameter learning when data sets are small and only one query is of interest.

That theory is worthless. It isn't even wrong!

– Wolfgang Pauli

To my Mother and Father

Acknowledgements

I would like to acknowledge the support and assistance of Russ Greiner (Dept. of Computing Science) and Peter Hooper (Dept. of Mathematical and Statistical Sciences) in the preparation of this thesis. I further acknowledge the financial support of the National Science and Engineering Research Council of Canada, the Alberta Informatics Circle of Excellence, and the University of Alberta.

Table of Contents

1 Introduction 1

2 Background 3
- 2.1 Bayesian Networks . . . 3
- 2.2 Dirichlet Distributions . . . 4
- 2.3 Interval Estimates . . . 5

3 Variance Propagation for Bayesian Error Bars 6
- 3.1 Introduction . . . 6
- 3.2 Problem Definition . . . 7
- 3.3 Derivation . . . 7
- 3.4 Efficient Calculation of Partial Derivatives . . . 9
  - 3.4.1 Bucket Elimination . . . 9
  - 3.4.2 Bucket Elimination Plus (BE^+) . . . 10
- 3.5 Bayesian Error Bars . . . 12
- 3.6 Experiments . . . 13
  - 3.6.1 The Normality Assumption in Practice . . . 13
  - 3.6.2 Accuracy of Error Bars . . . 14
- 3.7 Related Work . . . 15
- 3.8 Discussion . . . 16

4 Learning from Small Data Sets 17
- 4.1 Introduction . . . 17
- 4.2 Problem Definition . . . 17
- 4.3 Classical Techniques . . . 18
  - 4.3.1 Maximum Likelihood . . . 18
  - 4.3.2 Mean Posterior . . . 19
  - 4.3.3 Counting Statistics . . . 19
- 4.4 Related Work . . . 20
- 4.5 Proposed Methods . . . 20
  - 4.5.1 Bagging . . . 21
  - 4.5.2 Network Reduction . . . 21
  - 4.5.3 Finite Mixture Models . . . 22
- 4.6 Experiments . . . 23
  - 4.6.1 Experiment Design . . . 23
  - 4.6.2 Results . . . 23
- 4.7 Discussion . . . 25

5 Conclusion 29

Bibliography 30

A Multivariate Delta Rule 33

B Description of InfEB 35

C Variance of $q(\Theta)$ under Naïve Bayes 38

D Fisher Information on Bayesian Networks 40

List of Figures

2.1 Diamond network 4

3.1 InfEB algorithm 12
3.2 Alarm network: Quantile-Quantile Plots 14
3.3 Query sample coverage: 100 Alarm Queries 15

4.1 Two models: Inverted Naïve Bayes and Triangle 22
4.2 Experimental flow 24
4.3 Performance of MP, network reduction, and counting statistics: Inverted Naïve Bayes 25
4.4 Squared error of queries on Alarm, sample size 100 26
4.5 Squared error of queries on Alarm, sample size 400 27
4.6 Bayesian Data Score: two queries on Alarm 28

List of Symbols

$X_1 \dots X_n$	Variables or their corresponding nodes in a Bayesian network
$\|X_v\|$	Number of states random variable X_v can take
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	Bold font capital letter denote sets of variables
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Bold font lowercase letters denote configurations of variables
$\mathbf{F}_v$	Parents of a node/variable X_v in a Bayesian network
$D = \{d_1, \dots, d_n\}$	Data set. Each sample is denoted d_i
$Pr(\cdot)$	Probability distribution.
$Pr(\mathbf{Q} = \mathbf{q} \mid \mathbf{E} = \mathbf{e}) = Pr(\mathbf{q}\mid\mathbf{e})$	Value of a query response. The conditioning event is $\mathbf{E} = \mathbf{e}$.
$E[X]$	Expectation of X.
Θ	Network parameters. $\cup_v \cup_x \Theta_{v,x\mid\mathbf{f}}$
Θ^*	Expected network parameters. $\Theta^* = E[\Theta]$
$\hat{\Theta}$	Estimate of network parameters
$\Theta_{v,x\mid\mathbf{f}}$	Network parameter. Corresponds to $Pr(X_v = x \mid \mathbf{F}_v = \mathbf{f})$.
$\Theta_{v\mid\mathbf{f}}$	Distribution of X_v given $\mathbf{F}_v = \mathbf{f}$
$N_{v,x\mid\mathbf{f}}$	Number of cases in the data set where $X_v = x$ and $\mathbf{F}_v = \mathbf{f}$
$N_{v\mid\mathbf{f}}$	Number of cases in the data set where $\mathbf{F}_v = \mathbf{f}$. $N_{v\mid\mathbf{f}} = \Sigma_{i=1}^{r} N_{v,i\mid\mathbf{f}}$
$Dir(\alpha_{v,1\mid\mathbf{f}}, \dots, \alpha_{v,r\mid\mathbf{f}})$	Dirichlet distribution with r parameters
$\alpha_{v,x\mid\mathbf{f}}$	Dirichlet hyperparameter corresponding to $\Theta_{v,x\mid\mathbf{f}}$
$\alpha_{v\mid\mathbf{f}}$	Effective sample size of Dirichlet distribution. $\Sigma_{i=1}^{r} \alpha_{v,i\mid\mathbf{f}}$
$q(\Theta)$	Value of a query response. A function of network parameters Θ
σ_X^2	Variance of X
μ_X	Expected value of X
$\Sigma^{v\mid\mathbf{f}}$	Covariance matrix corresponding to row $v\mid\mathbf{f}$
$\frac{\partial q(\Theta^*)}{\partial \Theta_{v,x\mid\mathbf{f}}}$	Partial derivative of $q(\Theta)$ w.r.t. $\Theta_{v,x\mid\mathbf{f}}$ evaluated at $\Theta = \Theta^*$.
$b_0, b_1, \dots, b_n$	Sets of conditional probability tables (buckets).
$\Gamma(\cdot)$	Gamma function. $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$
$\|\cdot\|_2$	Euclidian (L_2) norm

Chapter 1

Introduction

Bayesian networks are probabilistic models that factor complicated probability distributions over many variables. A Bayesian network is composed of a directed acyclic graph that represents the qualitative dependency structure of the distribution and network parameters that define the distribution. The most common use of Bayesian networks is statistical inference, the calculation of a conditional probability (query response) of a configuration of variables given the observed state of other variables.

While networks can be constructed with parameters instantiated by human experts it is far more common to learn (estimate) from data. In this thesis, we concern ourselves primarily with issues stemming from parameter estimation in Bayesian networks. When data is abundant, it is easy to produce accurate estimates. When data sets are small, problems arise. The phrase "small data set" in machine learning is often a catchall explanation of why an algorithm did not work. Assuming the data is uncorrupted and the structure of the model correct, more data eventually leads to better estimators. It should be noted that smallness of a data set is relative. For example, estimating the mean of a univariate normal distribution requires fewer samples than its multivariate analogue. The former has fewer parameters than the latter; *ceteris parebus* it will require less data to produce an accurate estimate. Other factors, such as the underlying complexity of the distribution, play a role as well. In practice though, one is often presented with a single data set and no possibility for further samples.

This thesis examines issues that arise from limited data in two scenarios. The first problem involves calculating the variance of a query response. When network parameters are learned using data, a distribution over the parameters is formed to model parameter uncertainty. This in turn induces a distribution over a query response. When the data set is sufficiently large, the distribution of the query response is approximately normal. When the data set is small, we show that the Beta distribution provides a more accurate model of the query response.

The second problem considers parameter estimation when it is known that the estimator models all independencies between network variables. We consider the variant problem where the user cares only about one query response, and has reason to be concerned about the size of the data set.

Overview

Chapter 2 contains a short introduction to Bayesian networks, introducing both the notation and concepts required to understand this thesis. Estimation of the query response is covered in chapter 3. The mathematical foundations are discussed in section 3.3. The derivation of the algorithm used to estimate variance is contained in section 3.4, with details provided in appendix B. The method used to produce a model of the query response is described in section 3.5. The central experiments used to support our claim are found in section 3.6. Sections 3.7 and section 3.8 round out the discussion. Chapter 4 studies the single query small sample learning problem. A formal description of the problem and review of classical techniques are presented in sections 4.2 and 4.3 respectively. We consider related works and propose several algorithms for addressing this problem in sections 4.4 and 4.5. Experimental design and results are found in section 4.6. The virtues of the Beta model

of the query response are extolled in section 4.7, while the possibility for improvement in each algorithm is acknowledged. Appendix A describes the multivariate Delta rule, which is used in chapter 3. Appendix B describes an algorithm for computing the derivatives of a query with respect to network parameters. Appendix C studies the variance of queries on Naïve Bayes networks. A derivation of Fisher information for Bayesian networks is covered in Appendix D. Finally, we urge the reader to familiarize himself with the list of symbols in the front matter.

Chapter 2

Background

2.1 Bayesian Networks

Definition 2.1. *A Bayesian network $\langle \mathcal{V}, \mathcal{A}, \Theta \rangle$ encodes the joint distribution of a set of random variables $\mathbf{X} = \{X_1 \ldots X_n\}$. The graphical component is a directed acyclic graph (DAG) $\langle \mathcal{V}, \mathcal{A} \rangle$ which maps each node $v \in \mathcal{V}$ to a random variable $X_v \in \mathbf{X}$ and each directed arc $(u, v) \in \mathcal{A}$ to a dependency between variables X_u and X_v. Let $\mathbf{F}_v$ denote the variables whose nodes are parents of v. The parameters Θ consist of conditional probabilities $\Theta_{v,x|\mathbf{f}} = Pr(X_v = x|\mathbf{F}_v = \mathbf{f})$ which quantify the network.*

An example of a Bayesian network is presented in figure 2.1. Because of the correspondence between nodes and random variables, references to the graph are understood to be statements about the underlying random variables (and vice versa). Associated with each variable is a local function that maps an assignment of the parents to the conditional distribution of that variable. The joint distribution is simply the product of these local functions.

In practice the graphical component of a Bayesian network is sparse, only a small fraction of the possible arcs exist. Since the complexity of conditional probability distributions is largely dependent on the number of parents a variable has, sparseness allows for a compact *factored* representation of a joint probability distribution. Because the space requirement of a local conditional probability distribution is exponential in the number of parents, unfactored representations almost always require more space.

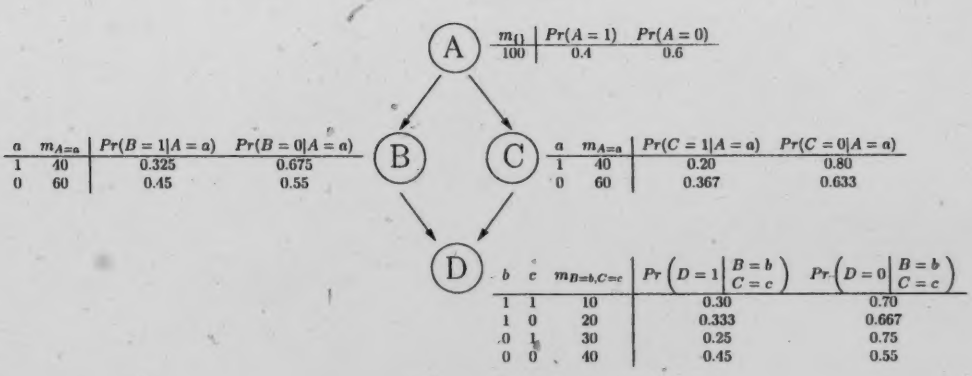
Bayesian networks are predicated on the notion that conditional independencies exist within the distribution being modelled. We introduce *I-maps* [40] to describe whether a particular directed acyclic graph captures all the dependencies in a distribution.

Definition 2.2. *An directed acyclic graph is an I-map of the distribution it models iff each random variable in the graph is independent of its non-descendants, given an assignment to its parents.*

Colloquially, a directed acyclic graph is an I-map if it is not missing any dependencies in the underlying distribution. However, I-maps may also contain extra dependencies, which do not exist in the distribution. The graph can still represent the distribution, but Θ will be larger than necessary. For example, a complete graph is always an I-map, but is space inefficient. A *minimal I-map* is one which contains no extraneous dependencies.

While Bayesian networks can be built on any combination of continuous and discrete variables, we restrict ourselves to networks containing only discrete variables with finite domains. Under this restriction the conditional probability distributions of a variable can be placed into a lookup table indexed by the conditioning event, which is an assignment to the variable's parents. We shall refer to these lookup tables as *CP-tables*. The conditional probability distribution of a variable that takes on r states will be denoted $\Theta_{v|\mathbf{f}} = (\Theta_{v,1|\mathbf{f}} \ldots \Theta_{v,r|\mathbf{f}})$.

Definition 2.3. *A query is the probability of a configuration of variables $\mathbf{Q} = \mathbf{q}$ given the configuration of variables $\mathbf{E} = \mathbf{e}$ where $\mathbf{Q}, \mathbf{E} \subseteq \mathbf{X}$ and $\mathbf{Q} \cap \mathbf{E} = \emptyset$. The value of probability*



$m_{\emptyset}$	$Pr(A=1)$	$Pr(A=0)$
100	0.4	0.6

a	$m_{A=a}$	$Pr(B=1\|A=a)$	$Pr(B=0\|A=a)$
1	40	0.325	0.675
0	60	0.45	0.55

a	$m_{A=a}$	$Pr(C=1\|A=a)$	$Pr(C=0\|A=a)$
1	40	0.20	0.80
0	60	0.367	0.633

b	c	$m_{B=b,C=c}$	$Pr\left(D=1 \middle\| \begin{matrix} B=b \\ C=c \end{matrix}\right)$	$Pr\left(D=0 \middle\| \begin{matrix} B=b \\ C=c \end{matrix}\right)$
1	1	10	0.30	0.70
1	0	20	0.333	0.667
0	1	30	0.25	0.75
0	0	40	0.45	0.55

Figure 2.1: Diamond network

$Pr(\mathbf{Q}=\mathbf{q}|\mathbf{E}=\mathbf{e})$ is sometimes called the query response. For a fixed network structure this value is dependent only on network parameters and is therefore denoted $q(\Theta)$.

The primary use of Bayesian networks is for *inference*, the calculation of query responses. Both exact and approximate inference on Bayesian networks is NP-hard [10, 11]. However, inference on Bayesian networks is almost always more efficient than marginalization of the unfactored joint distribution.

Thus far we have not considered how Bayesian networks are constructed. While both the DAG and the CP-tables may be constructed by human experts, it is far more common to induce a Bayesian network using data drawn from the distribution being modelled [9, 21, 27, 28, 44]. This is commonly called *learning*. Using data to construct the graphical component is referred to as *structural learning*. Estimation of the CP-tables is referred to as *parameter learning*. Throughout this thesis we presuppose knowledge of a minimal I-map, avoiding the need for structural learning.

A data set is a collection of independent and identically distributed assignments of the variables, random samples from the distribution being modelled. While both structural and parameter learning are possible given incomplete samples (*i.e.* samples where variable assignments are missing) [28] we restrict ourselves to the case of complete data.

There are two schools of thought on parameter learning – Bayesian and Frequentist. The former assumes that there is a prior distribution for each conditional probability distribution, and that data is integrated using Bayes' rule to derive a posterior conditional probability distribution. The latter does not assume the existence of a prior distribution[1]. In both cases learning is greatly simplified if we assume that the graphical structure partitions the data into multinomial samples [27]. That is, given an assignment to the parents of a variable, the samples whose assignments concur form a multinomial sample from the variable in question. In Frequentist parameter learning this allows conditional probability distributions to be learned using sample statistics. In Bayesian learning, if the prior is assumed to be Dirichlet distributed, data can be easily integrated using only sample statistics. This is because the Dirichlet distribution is the conjugate prior of the parameters for multinomial distributions [48].

2.2 Dirichlet Distributions

The Dirichlet distribution is a distribution over multinomial parameters. Thus they provide an ideal representation of parameter uncertainty. In the networks we consider, each CP-

[1]While these schools of thought appear throughout statistics, and are subject to endless philosophical debate and subdivision, the stance on prior distribution is a relatively uncontroversial way to differentiate them.

table row $\Theta_{v|\mathbf{f}}$ is modelled by a Dirichlet distribution with density

$$Pr(\Theta_{v,1|\mathbf{f}}, \ldots, \Theta_{v,r|\mathbf{f}}) = \frac{\Gamma(\alpha_{v|\mathbf{f}})}{\prod_{x=1}^{r} \Gamma(\alpha_{v,x|\mathbf{f}})} \prod_{i=1}^{r} (\Theta_{v,i|\mathbf{f}})^{\alpha_{v,i|\mathbf{f}}-1} \tag{2.1}$$

with mean and variance

$$E[\Theta_{v,x|\mathbf{f}}] = \frac{\alpha_{v,x|\mathbf{f}}}{\alpha_{v|\mathbf{f}}} \tag{2.2}$$

$$Var[\Theta_{v,x|\mathbf{f}}] = \frac{\alpha_{v,x|\mathbf{f}}(\alpha_{v|\mathbf{f}} - \alpha_{v,x|\mathbf{f}})}{(\alpha_{v|\mathbf{f}})^2(\alpha_{v|\mathbf{f}} + 1)} \tag{2.3}$$

However, inference algorithms require fixed network parameters. The common solution is to replace each $\Theta_{v|\mathbf{f}}$ with its expectation. As will be seen in chapter 3, this effectively ignores parameter uncertainty.

2.3 Interval Estimates

When the quantities being estimated are uncertain, providing a range to which the estimate likely belongs is common. In statistics, this is referred to as interval estimation. An $(1-\delta)$ interval estimate of parameter $\zeta \in \mathbb{R}^n$ is a compact region $\omega \in \mathbb{R}^n$ such that $Pr[\zeta \in \omega] = (1-\delta)$.

The semantics of intervals differs in the Frequentist and Bayesian world view. In Frequentist literature, ζ has a fixed unknown value. A $(1-\delta)$ interval means that $100(1-\delta)$ percent of intervals with coverage $(1-\delta)$ placed on the distribution will contain ζ. In Bayesian literature, ζ is a variable unknown. A $(1-\delta)$ interval is fixed with the belief that the probability ζ is in the interval is $(1-\delta)$. Because of the prior distribution in Bayesian procedures the two viewpoints produce different estimates given finite data. To differentiate the viewpoints we call interval estimates *credible regions* in Bayesian models and *confidence regions* in Frequentist models. We shall not consider confidence regions further.

Chapter 3

Variance Propagation for Bayesian Error Bars

3.1 Introduction

Bayesian network parameters can be elicited from domain experts. However, elicitation is time consuming and prone to logical inconsistency. Estimating network parameters using data provides an objective, and less arduous, alternative. Both procedures admit uncertainty in model parameters; but queries are often computed as if there is no uncertainty.

Even restricting our discussion to parameter estimates in Bayesian networks with discrete variables and the "correct structure", uncertainty is first modelled and then ignored. Uncertainty in local conditional probability distributions can be modelled using Dirichlet distributions [28]. However, inference algorithms assume that each network parameter is a single real value. Invariably the Dirichlet distributions are replaced with their expected values, thus dispensing with parameter uncertainty. Query responses, which are random variables, are replaced with point estimates.

The desirability of query variance in addition to a point estimate is readily apparent. Interval estimates (error bars) and techniques that require them become possible. In classification, high probabilities are frequently taken as a proxy for confidence. Error bars provide a more rigorous measure of confidence. In decision theory, variance can be used as a measure of risk. Applications of Bayesian networks to outlier detection could use query variance to differentiate sampling variation from true outliers [35].

An algorithm to estimate the first-order variance of a query response exists (section 3.4). Furthermore, it has been proven that $q(\Theta)$ is asymptotically normal [45, 47]. However, when the normality assumption is made, credible intervals are often inaccurate if $E[q(\Theta)]$ is near 0 or 1. Two causes have been posited for this behavior: the quality of the variance approximation and the normality assumption. This paper shall show that, in practice, the normality assumption is the primary source of error. Moreover, we propose the use of the Beta distribution to model $q(\Theta)$.

This chapter is structured as follows. The problem is formally introduced in section 3.2. Because of the subtlety of the question, theoretical foundations of variance estimation in Bayesian networks are covered in section 3.3. This result provides a first-order approximation; but the straightforward approach is impractically slow as it would require calculation of partial derivatives of the query response. Section 3.4 introduces BE^+, a relatively efficient algorithm for calculating the required partial derivatives. Distributional assumptions for the query response are also considered. Section 3.6 provides experimental support in favor of the Beta hypothesis. Section 3.7 is an overview of related literature. Section 3.8 concludes in favor of our thesis, discusses future extensions, and presents the reader with open questions.

3.2 Problem Definition

When a query response $Pr(\mathbf{q}|\mathbf{e})$ is computed on a network with no uncertainty, the response is a single number. When the same query is computed on a network with uncertainty, the response itself is uncertain and modelled by a distribution. The only source of variance we consider is that induced by parameter uncertainty. To emphasize the source of variation, the query response is sometimes denoted $q(\Theta)$ – a function of random variables Θ.

Throughout this chapter a Bayesian stance is taken. There is a prior distribution over parameters, which induces a prior distribution for $q(\Theta)$. Furthermore, we impose the following assumptions:

1. TRUE STRUCTURE: $\langle \mathcal{V}, \mathcal{A} \rangle$ is an I-map of the underlying joint distribution.
2. FINITE DOMAIN: Each variable has a finite number of values.
3. PARAMETER INDEPENDENCE: The distribution of a variable conditioned on one assignment to its parents is independent of the distribution conditioned on any other assignment to its parents. If $\mathbf{f}_1 \neq \mathbf{f}_2$ then $\Theta_{v|\mathbf{f}_1}$ and $\Theta_{v|\mathbf{f}_2}$ are independent (local parameter independence). Furthermore, the local conditional probability functions are independent (global parameter independence). Both requirements were first formalized in [44].
4. DIRICHLET ASSUMPTION: The distribution of a variable given an assignment to its parents is Dirichlet distributed.
5. FIXED POSTERIOR MEANS: The expected posterior parameters $\Theta_{v,x|\mathbf{f}}$ remain fixed strictly between 0 and 1.

These concur with the assumptions made for *mean posterior* (MP) learning of Bayesian network parameters. We note that assumption 4 implies that no parameter can be 0 or 1. The algorithms described herein can be applied to such scenarios, albeit at the loss of guaranteed asymptotic normality.

Two methods will be used to verify hypothesized distributions. By sampling each Dirichlet row distribution a sample from Θ is drawn, which induces a sample from $q(\Theta)$. Statistical tests exist to determine whether random deviates were generated by a particular distribution. A similar test draws a $(1-\delta)$ credible interval $[a, b]$. If the hypothesized distribution is correct one expects the fraction of samples that fall within this interval to be $(1-\delta)$. The validity of this test is a consequence of the Glivenko-Cantelli theorem[1] [12].

3.3 Derivation

In this section we apply the multivariate delta method [4] to derive an estimate of the variance of a query response. We introduce the following notation. Θ^* denotes the network parameters formed by replacing each Dirichlet row distribution with its expectation. $q'(\Theta^*)$ denotes vector of partial derivatives of scalar $q(\Theta^*)$ with respect to each $\Theta_{v,x|\mathbf{f}}$ and $q''(\Theta^*)$ denotes the matrix of second derivatives of $q(\Theta^*)$. Consider a first-order Taylor series expansion of $q(\Theta)$ around Θ^*:

$$q(\Theta) = q(\Theta^*) + q'(\Theta^*)[\Theta - \Theta^*]^T + R \tag{3.1}$$

$$R = \frac{1}{2}[\Theta - \Theta^*]q''(\Theta^*)[\Theta - \Theta^*]^T \quad \exists \Theta^* \in [\Theta^*, \Theta] \tag{3.2}$$

Since $q(\Theta)$ is twice-differentiable it can be shown (Appendix A) that

$$E[(q(\Theta) - \mu_{q(\Theta)})^2] \approx \hat{\sigma}^2_{q(\Theta)} = q'(\Theta^*) \cdot \Sigma \cdot q'(\Theta^*)^T \tag{3.3}$$

[1] *i.e.* the empirical distribution uniformly converges in δ as the number of samples from $q(\Theta)$ tends towards infinity. It also forms the basis for the Kolmogorov-Smirnov goodness-of-fit test.

This also requires the following result from [9]:

$$\mu_{q(\Theta)} = E[q(\Theta)] = q(E[\Theta]) = q(\Theta^*)$$

To compute a Bayesian credible interval around $q(\Theta^*)$ the cumulative distribution function of the query response is required. Theorem 1 argues that for sufficiently large sample size, a normal approximation is reasonable.

Theorem 1. *Under assumptions 1 through 5 as* $min\{\alpha_{v,x|\mathbf{f}}\} \rightarrow \infty$ *the standardized random variable*

$$\frac{q(\Theta) - E[q(\Theta)]}{\sigma_{q(\Theta)}}$$

converges in distribution to the standard normal distribution.

Proof. Under assumption 5 each Dirichlet row converges in law to a multivariate normal distribution [1]. Moreover, the remainder term in equation 3.1 is asymptotically negligible (all the second partial derivatives are bounded in a neighbourhood around Θ^*), ensuring that R converges to 0 more rapidly than the linear term. Thus, asymptotically, $q(\Theta)$ is a linear function of Gaussians and therefore itself Gaussian. □

Given that each CP-table row is an independent Dirichlet distribution the (i, j) element of covariance matrix Σ is non-zero iff Θ_i and Θ_j are from the same distribution. We can therefore decompose formula 3.3 into the sum of row contributions

$$\hat{\sigma}^2_{q(\Theta)} = \sum_{v|\mathbf{f}} q'_{v|\mathbf{f}}(\Theta^*) \cdot \Sigma^{v|\mathbf{f}} \cdot q'_{v|\mathbf{f}}(\Theta^*) \tag{3.4}$$

where $\Sigma^{v|\mathbf{f}}$ is the covariance matrix of $\Theta_{v|\mathbf{f}} = Dir(\alpha_{v,1|\mathbf{f}}, \ldots, \alpha_{v,r|\mathbf{f}})$ and

$$q'_{v|\mathbf{f}}(\Theta^*) = \left(\frac{\partial q(\Theta^*)}{\partial \Theta_{v,1|\mathbf{f}}} \cdots \frac{\partial q(\Theta^*)}{\partial \Theta_{v,r|\mathbf{f}}} \right)$$

noting from [24, 13] that

$$\frac{\partial q(\Theta)}{\partial \Theta^*_{v,x|\mathbf{f}}} = \frac{1}{\Theta^*_{v,x|\mathbf{f}}} \left[\begin{array}{l} Pr(\mathbf{Q} = \mathbf{q}, X_v = x, \mathbf{F}_v = \mathbf{f} | \mathbf{E} = \mathbf{e}) - \\ Pr(\mathbf{Q} = \mathbf{q} | \mathbf{E} = \mathbf{e}) \cdot Pr(X_v = x, \mathbf{F}_v = \mathbf{f} | \mathbf{E} = \mathbf{e}) \end{array} \right] \tag{3.5}$$

Theorem 2, described in [46], provides an alternate form of equation 3.4. Solving this equation requires only the ability to calculate arbitrary queries and the effective sample size of each Dirichlet row distribution, $\Theta_{v|\mathbf{f}}$.

Theorem 2. *Under assumptions 1-4 the approximate variance of query response* $q(\Theta) = Pr(\mathbf{q}|\mathbf{e})$ *is*

$$\hat{\sigma}^2_{q(\Theta)} = \sum_{v|\mathbf{f}} \underbrace{\frac{1}{\alpha_{v|\mathbf{f}} + 1} [A - B]}_{\text{variance of row } v|\mathbf{f}} \tag{3.6}$$

$$A = \sum_{v} \frac{1}{\Theta^*_{v|\mathbf{f}}} [Pr(v, \mathbf{f}, \mathbf{q}|\mathbf{e}) - Pr(\mathbf{q}|\mathbf{e}) Pr(v, \mathbf{f}|\mathbf{e})]^2 \tag{3.7}$$

$$B = [Pr(\mathbf{f}, \mathbf{q}|\mathbf{e}) - Pr(\mathbf{q}|\mathbf{e}) Pr(\mathbf{f}|\mathbf{e})]^2 \tag{3.8}$$

Proof. The covariance matrix of row $v|\mathbf{f}$ is

$$\Sigma_{ij}^{v|\mathbf{f}} = \begin{cases} \Theta^*_{v,i|\mathbf{f}}(1 - \Theta^*_{v,i|\mathbf{f}})/(\alpha_{v|\mathbf{f}} + 1), & i = j \\ -\Theta^*_{v,i|\mathbf{f}}\Theta^*_{v,j|\mathbf{f}}/(\alpha_{v|\mathbf{f}} + 1), & i \neq j \end{cases} \tag{3.9}$$

If we let I_n denote a $n \times n$ identity matrix then the covariance matrix can be factored as follows:

$$\Sigma^{v|\mathbf{f}} = \frac{1}{\alpha_{v|\mathbf{f}} + 1}\left[\Theta^*_{v|\mathbf{f}} \cdot I_v - \Theta^*_{v|\mathbf{f}} \cdot \left[\Theta^*_{v|\mathbf{f}}\right]^T\right] \tag{3.10}$$

Thus equation 3.4 becomes

$$\hat{\sigma}^2_{q(\Theta)} = \sum_{v|\mathbf{f}} \frac{1}{\alpha_{v|\mathbf{f}} + 1}\left[\sum_v q'_{v|\mathbf{f}}(\Theta^*)^2 \cdot \Theta^*_{v|\mathbf{f}} - \left(\sum_v q'_{v|\mathbf{f}}(\Theta^*) \cdot \Theta^*_{v|\mathbf{f}})^2\right)\right] \tag{3.11}$$

By substitution of the partial derivatives by their corresponding value in equation 3.5 and algebraic simplification we arrive at the required result. □

For specific graph structures, it may be possible to avoid computation of equation 3.6. Naïve Bayes is one such example (Appendix C). However, in general computing partial derivatives using equations 3.5 or 3.6 directly would be impractically slow.

3.4 Efficient Calculation of Partial Derivatives

In this section we describe the BE$^+$algorithm [45], which simultaneously computes $q(\Theta^*)$ and all the partial derivatives required for equation 3.3 in $O(n \cdot exp(w))$ time and space complexity, where n is the number of nodes in the network and w is the induced tree width of the variable ordering, π, used.

3.4.1 Bucket Elimination

BE$^+$extends the bucket elimination framework [15], a class of variable elimination algorithms that include methods for belief net inference. We introduce the following notation. Let $f(S)$ represent a function over a set of named variables $f : S \to \mathbb{R}$. That is, f maps each assignment $S = s$ to a real number. Bucket elimination is based upon two operators: *join* and *elim*. Join combines an arbitrary number of functions together to produce a new function

$$f(T) = \text{join}\{g_1(S_1), \ldots, g_r(S_r)\}$$

where $T = \bigcup_{i=1}^r S_i$ and

$$f(t) = \prod_{i=1}^r g_i(t)$$

It is understood that t is an assignment to the variables of T and $g_i(t)$ is the function when its arguments are set to their corresponding values in t, an assignment to a superset of the arguments of $g_i(\cdot)$. Elim marginalizes a set of variables from a function, producing a new function:

$$f(T) = \text{elim}_R[g(S)]$$

where $T = S - R$ and

$$f(t) = \sum_{r \in R} g(t : r)$$

It is understood that $t : r$ is an assignment to the variables in S using the corresponding assignments in t and r. Assignments to a subset of the arguments of a function are allowed. Instead of returning a single value, the result is a new function on the unassigned variables $t \in T$.

If we denote the set of CP-tables in a network $\{h_1(S_1), \ldots, h_n(S_n)\}$ and fix variables $\mathbf{E} = \mathbf{e}$ in all the table then

$$Pr(\mathbf{E} = \mathbf{e}) = \text{elim}_{\bigcup_{i=1}^{n} S_i}[\text{join}\{h_j(\mathbf{e}) \mid 1 \leq j \leq n\}]$$

However, direct evaluation of this expression is impractically slow and memory inefficient – it enumerates the full joint distribution (exponentially large in the number of variables) and then marginalizes. The key to bucket elimination is that it factors functions across the sums and products (elims and joins).

Given an ordering of variables $\pi = v_1 \ldots v_n$ bucket elimination creates a series of buckets $b_0, b_1, \ldots, b_n$, and for each evidenced CP-table $h_j(e)$ places it in the bucket of maximal variable. The buckets are processed in order $b_n, \ldots, b_1$. For each bucket b_i associated with variable X_i compute

$$f_j(T) = \text{elim}_{\{X_i\}}[\text{join}\{g_i(S_i) \mid g_i(S_i) \in b_i\}]$$

and place $f_j(T)$ in the bucket corresponding to the maximal variable (*i.e.* the variable of maximum index in the ordering). Degenerate functions, those lacking variables, are placed in b_0. The marginal probability $Pr(\mathbf{E} = \mathbf{e})$ is simply the product of functions in b_0.

3.4.2 Bucket Elimination Plus (BE$^+$)

Our description of BE$^+$, which computes both the query response and partial derivatives, is based upon a constructive proof from [45]. First, we simplify the problem with the following:

Lemma 1. *If a function $f(S)$ has a counterpart function $f^{-1}(S)$ such that*

$$y = Pr(\mathbf{q}|\mathbf{e}) = elim_S[join\{f(S), f^{-1}(S)\}]$$

then

$$f^{-1}(s) = \frac{\partial y}{\partial f(s)}$$

Proof. By definition

$$y = \sum_{s \in S} f(s) \cdot f^{-1}(s)$$

The result follows by simple differentiation. Note that the functions map to unknown real-valued parameters, which is why the derivative is not 0. □

Theorem 3. *Every function $f(S)$ placed in a bucket has a counterpart function $f^{-1}(S)$ such that*

$$y = elim_S[join\{f(S), f^{-1}(S)\}]$$

Moreover, $f^{-1}(S)$ can be computed using the potentials produced during bucket elimination.

Proof. We shall prove the result inductively.

BASE: Bucket b_0 contains only degenerate functions (real numbers) which when combined, produce y. Therefore for all functions $u \in b_0$

$$u^{-1} = \text{elim}_{\emptyset}[\text{join}\{\forall v \in b_0, v \neq u\}]$$

INDUCTIVE STEP: Consider an arbitrary bucket b_j where $f_j(T)$ is the result of processing b_j. By inductive assumption the counterpart function $f_j^{-1}(T)$ exists. Let the contents of b_j be denoted $\{g_1(M_1), \ldots, g_r(M_r)\}$. For any $g_i(M_i)$ we shall show its counterpart $g_i^{-1}(M_i)$ exists. Recall that

$$f_j(T) = \text{elim}_{\{X_j\}}[\text{join}\{g_1(M_1), \ldots, g_r(M_r)\}]$$

and also by assumption that

$$y = \text{elim}_T[\text{join}\{f_j(T), f_j^{-1}(T)\}]$$

Therefore

$$\begin{aligned} y &= \text{elim}_T[\text{join}\{f_j^{-1}(T), \text{elim}_{X_j}[\text{join}\{g_1(M_1), \ldots, g_r(M_r)\}]\}] \\ &= \sum_{t \in T} f_j^{-1}(t) \sum_{x \in X_j} \prod_{j=1}^{r} g_j(t:x) \\ &= \sum_{\langle t,x \rangle \in T \times X_j} f_j^{-1}(t) \prod_{j=1}^{r} g_j(t:x) \end{aligned}$$

Since $M_i \subset T \cup X_j$ define $S = T \cup X_j - M_i$. Consequently

$$\begin{aligned} y &= \sum_{m \in M_i} g_i(m_i) \sum_{s \in S} f^{-1}(m_i:s) \prod_{j \neq i} g_j(m_i:s) \\ &= \text{elim}_{M_i}[g_i(M_i), \text{elim}_S[\text{join}\{\{f_j^{-1}(T)\} \cup \{g_j(M_j) \mid 1 \leq j \leq r, j \neq i\}\}]] \end{aligned}$$

Finally

$$g_i^{-1}(M_i) = \text{elim}_S[\text{join}\{\{f_j^{-1}(T)\} \cup \{g_j(M_j) \mid 1 \leq j \leq r, j \neq i\}\}]$$

□

The proof is constructive. After processing the buckets under a given ordering to calculate $Pr(\mathbf{E} = \mathbf{e})$ or $Pr(\mathbf{Q} = \mathbf{q}, \mathbf{E} = \mathbf{e})$ a second pass in reverse order computes the partial derivatives of a marginal with respect to each CP-table. Given the partial derivatives, computing equation 3.4 is straightforward.

Note that BE$^+$refers specifically to the algorithm used to calculate partial derivatives. The entire process will be called InfEB, and is illustrated in figure 3.1. Before the process begins we assume that uncertain network parameters Θ have been given to us as Dirichlet rows. Steps 1 and 2 illustrate this as mean posterior parameter estimation. It should be noted that InfEB also applies to networks whose parameters were constructed by maximum likelihood (as long as all the parameters are specified). In step 3 the posterior network and query are used to initialize data structures and run BE$^+$. Posterior parameters are replaced with their expectations to produce a mean network. The parameters of the mean network are denoted Θ^*. The forward pass of bucket elimination on the mean network outputs the expected marginal; the reverse pass produces the partial derivatives of the marginal evaluated at Θ^*. Thus the expected marginals $Pr_{\Theta^*}(\mathbf{e})$ and $Pr_{\Theta^*}(\mathbf{q}, \mathbf{e})$ as well as the partial derivatives (step 4) are produced. The outputs of BE$^+$are used in step 5 to calculate equation 3.4. This yields the variance of the query response. Additionally, the mean and variance of the query response can be used to produce a model of the distribution. Given a coverage probability $\delta \in (0, 1]$ the quantile function of the normal or Beta distribution is used to produce an approximate credible interval $[a, b]$ on $q(\Theta)$.

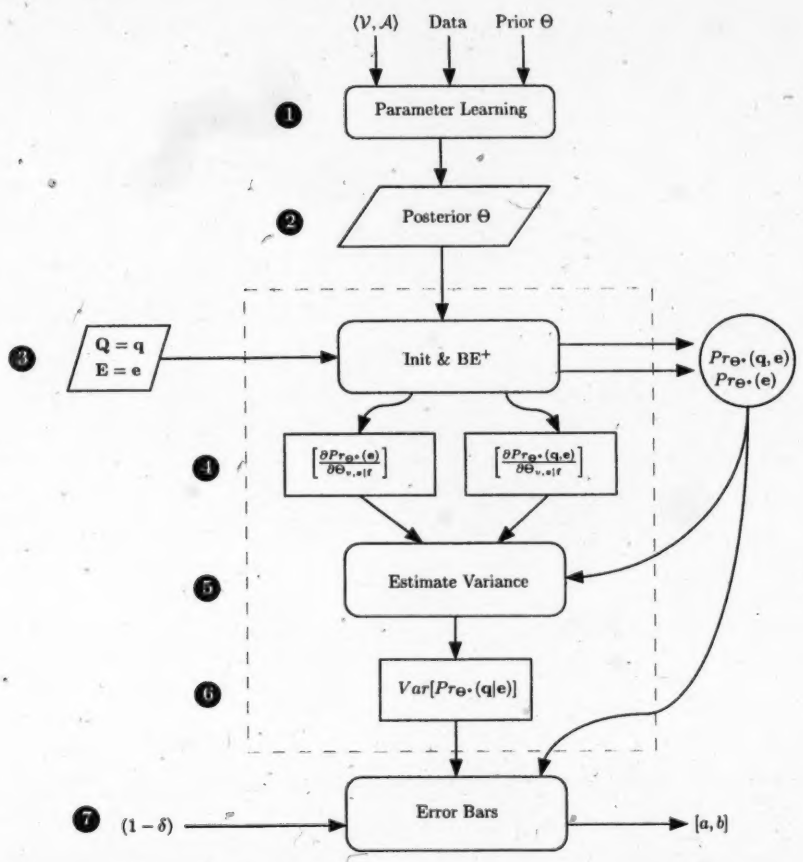


Figure 3.1: The flow of the algorithm used for estimating the variance of query response. The steps outside the dotted box are not part of InfEB.

3.5 Bayesian Error Bars

The outputs of BE$^+$are the first and second moments of the query response. Using the normal approximation advocated by theorem 1 is straightforward. Fitting a normal distribution to the query response given the calculated mean and variance is done using the method of moments [48]. Approximate credible intervals are formed by taking confidence intervals for the fitted normal distribution. Formally for any $\delta \in (0, 1]$ an approximate credible interval of coverage probability $(1 - \delta)$ centered at $q(\Theta^*)$ is defined as

$$[q(\Theta^*) - \epsilon,\ q(\Theta^*) + \epsilon]$$
$$\text{where} \quad \epsilon = -\Phi^{-1}(\delta/2) \cdot \hat{\sigma}_{q(\Theta)}$$

and $\Phi^{-1}(\cdot)$ is the inverse cumulative distribution function of a standard normal variable.

In practice, it is often dangerous to assume that the query response is normally distributed. While the query response is asymptotically normal, no claim is made about the rate of convergence. It has been observed in [47] that, on the Diamond network with effective sample size 10, the samples of several query responses deviated significantly from normality. The normal distribution is defined on $\mathbb{R}$, while the query response is by definition restricted to the [0,1] interval. In many cases significant probability mass is placed outside the unit interval. Moreover, the empirical distribution of the query response is often

skewed when the expected response is near 0 or 1. This cannot be accurately modelled by a normal distribution. We therefore propose the use of the Beta distribution to model $q(\Theta)$. It can take on positive, negative, or zero skew. Its parameters are also readily estimated using the method of moments. Finally, the normalized Beta distribution converges in law to the standard normal distribution. For sufficiently large sample sizes, it will produce results close to the asymptotic behaviour of $q(\Theta)$.

3.6 Experiments

We claim that Beta distribution is a better model of the query response than the normal distribution. To validate this claim we use a simple Monte Carlo strategy to sample from the distribution of the query response. Generate r replicates from Θ, denoted $\{\Theta^i\}_{i=1}^r$. Each replicate instantiates a Bayesian network with fixed parameter values. For each instantiated network calculate $Q_i = q(\Theta^i)$ using any algorithm for Bayesian network inference. These $\{Q_i\}$ are samples from the true distribution of the query response. Our experiments are based on $r = 1000$ samples.

Two network topologies are used in our experiments: Diamond and Alarm. The former, illustrated in figure 2.1, is a small network that allows for a variety of inferential patterns. The latter, described in [29], was designed by medical experts for monitoring intensive care patients. Since Alarm is not specified with Dirichlet rows, and often contains probabilities that are 0 or 1, we use our own parameters instead. We gave both networks an effective sample size of 100.

Using the Diamond network the following queries are studied:

1. $Pr(A = 1)$
2. $Pr(A = 1|B = 1)$
3. $Pr(A = 1|B = 1, C = 1)$
4. $Pr(B = 1, C = 1|A = 1)$
5. $Pr(A = 1|D = 1)$
6. $Pr(D = 1|A = 1)$

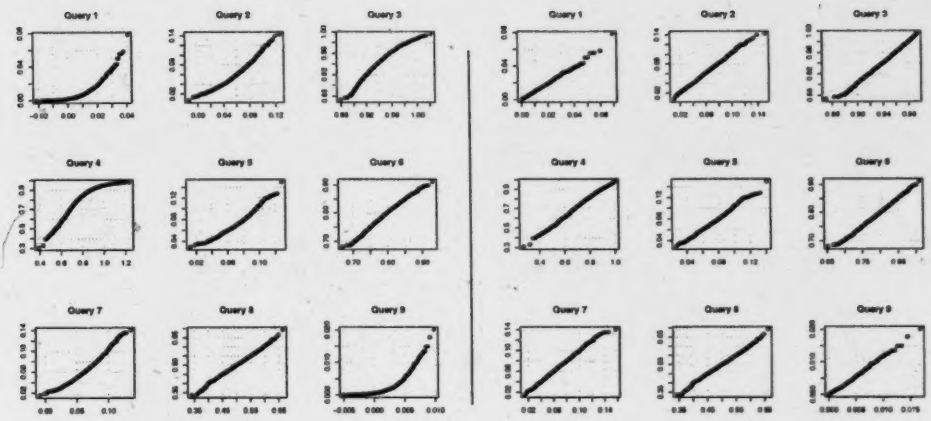
Using the Alarm network, 100 queries were generated by choosing a single query variable and three to five evidence assignments (using [29] to determine which variables could be query variables, and which could be evidence variables).

For a given network, the samples $\{Q_i\}_{i=1}^r$ are used to answer two questions. One, while $q(\Theta)$ is asymptotically normal, is it reasonable to assume normality in practice ? Two, how accurate are the error bars produced by our algorithm ?

3.6.1 The Normality Assumption in Practice

It was observed that queries on the Diamond network with effective sample size 10 exhibit significant non-normality. We formally define significant deviation from normality as a p-value of less than 0.01 on a Shapiro-Wilk test (*i.e.* the probability that the data is drawn from a normal distribution with population mean and variance is less than 0.01). Even if the effective sample size is increased tenfold, queries 2, 3, and 4 continue to deviate significantly from normality. For the Alarm query responses, 85 deviated significantly from normality. The problem is especially pronounced when $q(\Theta^*)$ is near 0 or 1.

When the normal distribution is replaced by a Beta distribution, as discussed in section 3.5, the results improve tremendously. The Shapiro-Wilk test, which applies only to normality testing, cannot be applied here. The Kolmorogov-Smirnov test [12] is a goodness-of-fit test that can be used with arbitrary distributions. It should be noted that when the Kolmorogov-Smirnov test is usually used, only the sample moments are known. The test is conservative in this situation. Since we are using the true mean and an estimate of the variance not dependent on the samples, the test is accurate. A stringent significance threshold of 0.001 is used (*i.e.* if the p-value is less than 0.001, the data is deemed not to



(a) Normal Hypothesis

(b) Beta Hypothesis

Figure 3.2: Alarm network: Quantile-quantile plots comparing query samples against estimates of $q(\Theta)$ using calculated mean and variance. The line $y = x$ indicates perfect concordance between the data and proposed distribution. If the samples deviate substantially from this line the data does not accord with the proposed distribution.

conform to the hypothesized distribution). We evaluated the 100 Alarm queries: 58 conformed to the normal model; 96 conformed to the Beta model.

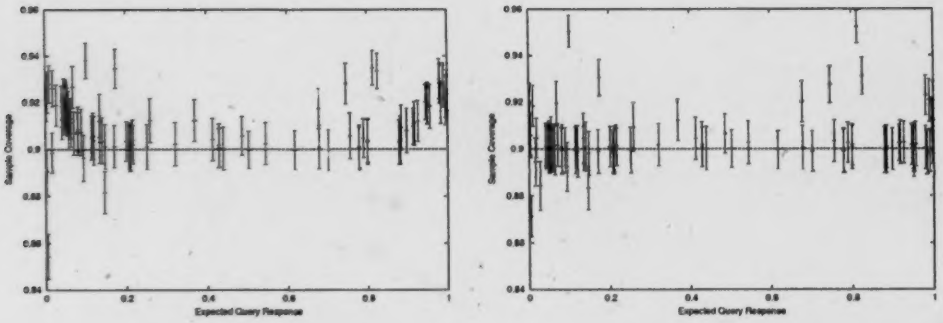
Quantile-quantile plots provide an anecdotal, but more persuasive illustration of our thesis. Sample quantiles are plotted against the theoretical quantiles of the fitted normal and Beta distributions. In figure 3.2 nine representative queries were chosen to illustrate the quality of the normal approximation. When the true distribution is skewed the failure of the normal model is most prominent. The Beta distribution appears to model all the queries at least as well as the normal distribution, and often much better.

3.6.2 Accuracy of Error Bars

A common use of variance is to produce interval estimates: confidence regions in the Frequentist framework and credible regions in the Bayesian framework. Since BE$^+$is predicated on a Bayesian stance we concern ourselves only with credible intervals.

We restrict our examination to the Alarm network. The 100 queries chosen contain results with both high and low variance, as well as many queries with expected response near 0 or 1. A wide variety of inferential patterns can be observed on one network. Similar behavior was observed with the Diamond network; the results are elided for brevity. For each query a posterior coverage experiment is performed. A normal or Beta distribution is constructed as per section 3.5. We then approximate a 90% credible interval and record the fraction of query samples $\{Q_i\}$ in the interval. To address sampling variation, the procedure is repeated 100 times for each query. Results are presented in figure 3.3.

Assuming the query response is normally distributed tended to produce overly large intervals on Alarm. This may be due to higher kurtosis of the normal distribution compared to the beta distribution. However, we cannot claim that the algorithm is always conservative. On a different network, interval estimates could be systematically liberal. On the other hand, the Beta distribution tends to produce more accurate intervals. It should be remembered that the variance estimate is only a first-order approximation. For certain queries, the remainder term (R from equation 3.1) can still have a noticeable effect. A few of the outliers also correspond to cases where the Beta distribution is degenerate (*e.g.* U-shaped).



(a) Normal Hypothesis

(b) Beta Hypothesis

Figure 3.3: Query sample coverage of computed 90% Bayesian credible intervals. Each point represents a query on Alarm. On the left we assume $q(\Theta)$ is normally distributed. On the right we assume $q(\Theta)$ is Beta distributed. The points and 1σ error bars represent the sample mean and deviation over 100 coverage experiments. The line represents the desired result, 90% coverage.

In such scenarios neither the normal nor the Beta model is appropriate. These cases can be marked by looking at the hyperparameters of the Beta model.

It should also be noted that the presentation of results in figure 3.3 helps explain certain anomalies found in [45]. In that work, the error in empirical coverage of an interval is averaged over 30 networks, formed using different data sets of size m. Failure to report the variance in coverage errors due to training set variation could mislead the reader to conclude that for certain queries, the algorithm becomes less accurate as m increases.

3.7 Related Work

This paper continues the work of [47] by proposing an alternative to the Gaussian approximation of the query response. The algorithm itself essentially propagates uncertainty in CP-table rows through a system of partial derivatives. While the mathematical foundations are a straightforward application of techniques from error analysis, the application to Bayesian networks has not been widely explored.

Kleiter [31] is closest in motivation to this work. It presents a framework for estimating query variance given complete or missing data. However, it (1) assumes all variables in a Bayesian network are independent, (2) ignores correlations within rows, (3) does not take advantage of local parameter independence, and (4) is dependent on stochastic simulation to approximate its variance estimate. Moreover, our results present empirical support with regards to the accuracy of variance estimates.

A larger body of related work considers *sensitivity analysis*, exploring how sensitive network parameters and queries are to changes in other network parameters [19, 43, 7]. These papers do not consider variance, but sometimes assume that parameters are intervals to be propagated throughout the network. In contrast, our assumption of independent Dirichlet rows is in accord with common MP models of learning Bayesian network parameters. We explain the source of the intervals, instead of treating them as user-defined ranges.

Of especial note are Query-DAGs [13], which provide an alternate algorithm for computing the first partial derivatives of a query with respect to network parameters. While asymptotically no faster than BE$^+$, in practice it avoids the creation of intermediate CP-tables, a bottleneck in our implementation.

Throughout our experiments we used samples from $q(\Theta)$. The samples themselves can be used to draw approximate credible intervals. This technique is computationally expensive, requiring an exponential time algorithm to produce each sample. There are applications of sampling methods to other problems in Bayesian networks which can easily be confused with our concerns. In stochastic sampling inference algorithms, confidence intervals on the posterior [8] refer to the distribution induced by sampling. The underlying network has no parameter uncertainty.

3.8 Discussion

In this chapter we put forward the thesis that the Beta distribution provides a better model for $q(\Theta)$ than the Gaussian model. This has been shown to be the case on a wide variety of inferential patterns by both theoretical measures (goodness-of-fit tests) and applications (posterior coverage experiments). Furthermore, our results reinforce the argument that a first-order variance approximation is almost always sufficiently accurate.

While our current system takes advantage of many optimizations available to bucket elimination algorithms (*e.g.* optimized variable orderings), it would be desirable to use a multiple query algorithm to amortize memory allocations over many queries. Query-DAGs [14], given their similarity to bucket elimination, are an obvious choice.

Another possible criticism is that missing data is not handled. The problem lies in overestimation of the effective sample size by the EM algorithm of [32]. Solving this problem immediately allows one to use the methods of this chapter to estimate the variance of query response. A related stance criticizes the assumption of discrete variables. However, many alternative representations (*e.g.* Noisy-OR [40], CP-Trees [5]) do not encode parameter uncertainty. Gaussian networks [22] are a notable exception. Local conditional probability distributions are represented as normal distributions. The joint distribution is thus multivariate Gaussian. The covariance matrix of the joint distribution is readily generated, but no consideration has been given to the question of the distribution of a query response.

One promising direction of research is the development of specialized versions of equation 3.3 that, for specific types of structures, bypass BE^+altogether. We have derived a closed form solution for the variance of query response in Naïve Bayes networks (Appendix C). We further believe that a similar derivation is possible for the second partial derivatives, allowing the error term of equation 3.1 to be easily calculated.

Does the query response belong to a single parametric family, regardless of the network and query ? The results of the chapter suggest that the Beta distribution is a viable candidate. Alternately, one can take a less stringent approach and explore richer representations of $q(\Theta)$. Mixture models or Bernstein polynomials of Beta distributions [23] are two possible directions, albeit both bearing the cost of sampling.

Chapter 4

Learning from Small Data Sets

4.1 Introduction

The estimation of Bayesian network parameters is a frequently addressed problem [9, 21, 28, 44]. This is especially true for discrete variable networks and complete data sets. However, relatively little attention has been given to parameter estimation when the data sets are small and only one query is of interest[1]. In this chapter we examine this issue. Section 4.2 defines the problem. Section 4.3 introduces classical approaches to parameter estimation in Bayesian networks. Section 4.4 discusses related work, none of which quite address this problem. Section 4.5 proposes three approaches to small sample estimation of the query response. Section 4.6 describes the experimental design, and presents the results. Discussion, including an argument on how algorithms should be compared, is presented in section 4.7.

4.2 Problem Definition

Assume a fixed distribution $\mathcal{P}$ can be modelled by a minimal I-map $\langle \mathcal{V}, \mathcal{A} \rangle$ and parameters Θ^*. That is, the distribution of interest can be modelled by a Bayesian network. Given the following:

1. *True Structure*: A minimal I-map of $\mathcal{P}$
2. *Complete, Noiseless Data*: A small finite data set drawn from the fixed distribution $\mathcal{P}$, without missing or incorrect variable assignments.
3. *Explicit Conditional Distributions*: Each conditional probability distribution must be either a multinomial (in a Frequentist framework) or Dirichlet (in a Bayesian framework) distribution[2]
4. *Single Query*: There is only one query of interest. The probability of this query is known as the response, $q(\Theta)$, a function on the unknown network parameters Θ. The true query response is denoted $q(\Theta^*)$.

induce a parameter estimate $\hat{\Theta}$ that minimizes L_2 error with respect to the query of interest:

$$err_{q(\Theta)} = \| q(\Theta^*) - q(\hat{\Theta}) \|_2 = [q(\Theta^*) - q(\hat{\Theta})]^2 \tag{4.1}$$

[1] This is not inductive learning, where the goal is to calculate $Pr(\mathbf{q}|\mathbf{e})$ for different sets of observations $\mathbf{E} = \mathbf{e}$. Rather we fix the observations and consider a particular set of query variables.

[2] This is often called the *unrestricted multinomial model* for conditional probability distributions [28]. The term refers to the way data is generated, but becomes confusing when reference is made to Bayesian methods which use a Dirichlet distribution to estimate the parameters of an unrestricted multinomial model.

If two estimators are compared, the one with lower L_2 error is referred to as more accurate. As the size of the data set tends towards infinity, both the maximum likelihood and mean posterior estimators will produce arbitrarily accurate estimates. However, in practice, data sets are often too small to produce very accurate estimators. Another consequence of small data sets is the smoothing problem. Maximum likelihood estimators do not assume a prior. Without data matching the conditioning event, no estimate can be produced. This frequently occurs for conditional probability distributions with many conditioning events. Only samples whose assignments match the conditioning events can be used to estimate the distribution.

The simplicity of the problem also allows us to address the question of training sample variation. The normative practice in the machine learning community is to use a single training set to construct a classifer. Either the entire training set is used or a subset is withheld to assess generalization error. Neither method takes into account the variation *between* training sets. It is obvious that different training sets of the same size can produce radically different estimates of Θ (and thus $q(\Theta)$) – especially if the training sets are small. Our goal is to determine which of the proposed algorithms is best suited for this problem. One would expect a superior algorithm to produce more accurate estimators given a variety of training sets.

4.3 Classical Techniques

Given the graphical component of a Bayesian network there are two common techniques for estimating Θ: maximum likelihood (ML) and mean posterior (MP). Both are used throughout statistics. A more general discussion of ML and MP can be found in [48]. An understanding of the following sections is requisite to later discussions in this chapter.

4.3.1 Maximum Likelihood

Given a data set D of independent samples and a density function of known form with unknown parameter ζ the principle of maximum likelihood argues that one should maximize the likelihood of the data given ζ. The same principle applies to vectors of unknown parameters.

For our purposes the likelihood function is simply $Pr(D|\Theta)$, or equivalently $\log Pr(D|\Theta)$. Let $N_{v,x|\mathbf{f}}$ be the number of samples that correspond to X_v taking on value x given an assignment $\mathbf{f}$ to the parents of X_v.

$$\begin{aligned}
\hat{\Theta} &= \underset{\Theta}{\operatorname{argmax}} \left[\log Pr(D|\Theta)\right] \\
&= \underset{\Theta}{\operatorname{argmax}} \left[\log \prod_{v|\mathbf{f}} \prod_{x} \Theta_{v,x|\mathbf{f}}^{N_{v,x|\mathbf{f}}}\right] \\
&= \underset{\Theta}{\operatorname{argmax}} \left[\sum_{v|\mathbf{f}} \sum_{x} N_{v,x|\mathbf{f}} \log \Theta_{v,x|\mathbf{f}}\right]
\end{aligned}$$

This is equivalent to maximizing the innermost sum for each conditional probability distribution. Because $\Theta_{v|\mathbf{f}}$ is a probability distribution the constraint $\sum_x \Theta_{v,x|\mathbf{f}} = 1$ applies. The maximum likelihood solution for each parameter is as follows:

$$\hat{\Theta}_{v,x|\mathbf{f}} = \frac{N_{v,x|\mathbf{f}}}{\sum_x N_{v,x|\mathbf{f}}}$$

If no samples match the conditioning event $\mathbf{F}_v = \mathbf{f}$ then the estimator is undefined. If $q(\Theta)$ is a one-to-one function the *invariance property* implies that $q(\hat{\Theta})$ is also a maximum

likelihood estimator [48]. In general, this does not hold for $q(\Theta)$. While not the goal of this chapter, a novel contribution of this work is the ability to calculate the covariance matrix of the generative distribution under maximum likelihood (Appendix D).

4.3.2 Mean Posterior

Just as with maximum likelihood, we are presented with a data set D of independent samples and a density function of known form and unknown parameter(s) ζ. The mean posterior approach is Bayesian. A prior over the parameters is established. Data is integrated to derive a posterior distribution. The mean of the posterior distribution constitutes our estimate. Unlike maximum likelihood, we are forced to make an assumption about the prior distribution over parameters, $Pr(\Theta)$. Without domain knowledge, it is common to choose a uniform distribution. For parameter estimation in Bayesian networks local parameter independence [44] allows this prior to be factored over each conditional probability distribution $\Theta_{v|\mathbf{f}}$.

Calculating the posterior distribution $Pr(\Theta|D)$ is a process of integrating data into the prior distribution. For arbitrary types of prior distributions, this step is computationally infeasible. However, we have assumed the data is multinomially distributed. By [48, 28] if the prior is assumed Dirichlet distributed, then integrating multinomial data yields a Dirichlet posterior. Moreover, integration requires only the computation of counts over the data set. This is called the *conjugate prior* relationship between multinomial and Dirichlet distributions.

One advantage of this technique over maximum likelihood is that parameters are always defined, even in the absence of data. The uniform prior is numerically identical to *Laplacian smoothing* of parameter estimates:

$$\hat{\Theta}_{v,x|\mathbf{f}} = \frac{N_{v,x|\mathbf{f}} + 1}{\sum_x N_{v,x|\mathbf{f}} + |X_v|}$$

which is equivalent to modeling each row as an average of the maximum likelihood estimate and the uniform prior mean [41].

4.3.3 Counting Statistics

Given a query $Pr(\mathbf{q}|\mathbf{e})$ the required marginals $Pr(\mathbf{q}, \mathbf{e})$ and $Pr(\mathbf{e})$ can be approximated using their empirical distributions on the data set. That is, the number of samples matching $\mathbf{Q} = \mathbf{q} \wedge \mathbf{E} = \mathbf{e}$ divided by the number of samples matching $\mathbf{E} = \mathbf{e}$. If we assume $Pr(\mathbf{Q}|\mathbf{e})$ is a multinomial distribution the problem can be viewed as maximum likelihood estimation. Likewise, if we assume a Dirichlet prior parameter learning can be viewed as mean posterior estimation.

Computation of the variance of query response is trivial. Given n tuples which match the conditioning event apply the multinomial variance.

$$\frac{Pr(\mathbf{q}|\mathbf{e})[1 - Pr(\mathbf{q}|\mathbf{e})]}{n}$$

for the maximum likelihood model or

$$\frac{E[Pr(\mathbf{q}|\mathbf{e})](1 - E[Pr(\mathbf{q}|\mathbf{e})])}{\alpha + 1}$$

where α is the effective sample size of the Dirichlet distribution on $Pr(\mathbf{Q}|\mathbf{e})$.

This is a poor algorithm if the data sets are small. The variance of the distribution can be high (or infinite in the case of maximum likelihood). This happens frequently if the query involves many conditioning events. However, the algorithm becomes attractive in certain scenarios. For example, when the query $Pr(C)$ is required and the network has an inverted Naïve Bayes topology (figure 4.1(a)).

4.4 Related Work

A natural formulation of our problem is in the discriminative learning framework. Generative techniques like ML and MP maximize a log-likelihood function on the data to find parameters that best model the joint distribution. The discriminative formulation maximizes a conditional log-likelihood function on the data to find parameters that best model one conditional distribution, the query of interest. This is equivalent to finding a model that minimizes the conditional cross-entropy of the query response [21]. While generative formulations are criticized for being indirect, optimizing the joint distribution when we often care only about one query, there are two limitations to the discriminative version. Tractable procedures for discriminative parameter estimation are not known to exist [26, 39]. Furthermore, the second reference strongly suggests that while discriminative parameter estimation leads to lower asymptotic error; the generative formulation asymptotes with fewer samples. The high sample complexity of the discriminative algorithm leads to lower accuracy than MP. For this reason, we focus on improving generative learning algorithms.

Other approaches simply ignore the I-map altogether. More generously, these algorithms do not presume foreknowledge of an I-map. If one is working with small amounts of data an approximate structure with fewer parameters often produces more accurate estimates of a query response than the true structure. This argument underlies the ubiquity of Naïve Bayes [33] and the promise of tree-augmented networks [21].

Ignoring the I-map is a waste of useful information. A less radical alternative is to replace CP-tables with approximate representations. Such techniques reparameterize Θ while respecting known dependencies. The oldest reparameterization technique, Noisy-OR [30, 40], predates learning algorithms for Bayesian networks. Noisy-OR is limited to distributions with binary variables. It treats each node like an logical-OR gate with its parents as inputs. However, there is a non-zero probability that when one parent is set to 1, the node will not output 1. Each such probability is independent of the state of all other parents. Finally, an extra "leak" node L is often added to account for exogenous effects:

$$Pr(X_v = 1|\mathbf{F}_v = f) = 1 - \left[Pr(X_v = 0|L) \prod_{F_j \in \mathbf{F}_v, F_j = 1} Pr(X_v = 0|F_j = 1) \right]$$

CP-table representations require space exponential in the number of parents; Noisy-OR representations require space linear in the number of parents. Under the assumption of independent causes other logical gates can be used similarly. Sigmoidal networks [38] represent conditional probabilities as sigmoid functions of weighted inputs from independent parent nodes. Conditional probability trees [5] exploits dependencies between CP-table rows. A CP-table can be viewed as a tree where the inner nodes contain assignments to a subset of variables $\mathbf{F}_v$. The value $Pr(X_v = x|\mathbf{F}_v = \mathbf{f})$ is on the leaf. CP-trees replace subtrees with a leaf when the loss of accuracy in the CP-table representation is minimal. This is similar to pruning in decision trees.

The discriminative model assumes only one class of queries $Pr(\mathbf{Q}|\mathbf{e})$ is of interest. What if there is a distribution of queries ? What parameters maximize performance across the query distribution ? In [24] it is shown that under the error function

$$err[q(\hat{\Theta})] = E_{q(\Theta)}[q(\Theta) - q(\hat{\Theta})]^2$$

approximation of optimal parameters is NP-Hard. This does not imply that our problem, having a degenerate query distribution, is also NP-Hard. Another difference is our concern with small data sets.

4.5 Proposed Methods

In this section we propose methods that address the problems exhibited by classical techniques: sample complexity, smoothing, and training set variance.

4.5.1 Bagging

Bootstrap aggregation, or bagging [6], uses a single data set to produce multiple estimators and then aggregates to output a single estimate. This is done through the creation of multiple data sets by a process known as bootstrapping [18]. Given a fixed data set D, another equally sized data set is produced by sampling cases from D with replacement. When the bootstrapped data set is smaller than D, the bagging algorithm is referred to as m-bagging. Each data set is independently used to produce an estimate. Aggregation usually consists of averaging estimators for regression or voting schemes, such as plurality, for classification. Since $q(\Theta)$ is a real-valued quantity we use averaging.

Formally let D_i represent one of the n^n possible data sets generated by bootstrapping. For any estimator of a real-valued parameter ζ the bagged estimate is as follows:

$$\zeta^B(D) = \sum_{i=1}^{n^n} Pr(D_i) \cdot \zeta(D_i)$$

Since this is computationally infeasible for even relatively small data sets approximate integration using $t \ll n^n$ must suffice:

$$\zeta^{\hat{B}}(D) = \frac{1}{t} \sum_{i=1}^{t} \zeta(D_i)$$

Our motivation for using bagging is straightforward: it tends to lower variance at the cost of higher bias [3]. Small data sets lead to high variance estimators of $q(\Theta^*)$. The tradeoff should lead to lower overall L_2 error. Furthermore, while bagging can make poor estimators far worse, because of the effect small changes have near the decision boundary, such criticisms have not been observed in regression.

A salient criticism of bagging is computational cost. This is especially true when the underlying estimation procedure $\zeta(D_i)$ is costly. The additional cost of generating data is nominal. In our experiments, the underlying routine is MP, which for complete data is quite efficient[3]. Finally, parallelization of bagging algorithms is trivial, requiring communication only to dispatch data sets and aggregate estimates.

4.5.2 Network Reduction

Standard inference algorithms effectively operate through the marginalization of variables from the network. It may be the case that variables (and their associated arcs) can be similarly removed, creating a new model with fewer parameters to estimate. Herein lies the motivation for a technique known as *network reduction*.

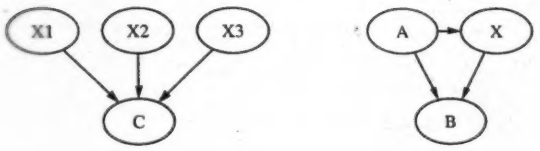
Our focus on network reduction will be on a hitherto unverified approach described in [25]. It addresses the problem of section 4.2, but assumes "a fixed set of independency claims" instead of a minimal I-map. While I-mapness is a property of directed acyclic graphs, independency claims can be made with respect to any graphical model. However, the examples used to motivate the paper are all I-maps – the difference is moot.

The technique allows one to remove a set of nodes Y from the network, producing a reduced graph. The variables associated with Y are removed from the data set, producing a reduced data set, until ML can be used to estimate all the parameters of the reduced network, Θ'. Finally, an estimate of the query response is produced by inference on the reduced network. Reduction is subject to three constraints:

1. For each $\Theta'_{v,x|\mathbf{f}}$ in the reduced graph there is at least one data sample matching the conditioning event $\mathbf{F}_v = \mathbf{f}$.

2. Evidence nodes that influence the query in the original graph must continue to do so in the reduced graph.

[3]This is especially true with the introduction of more efficient data structures for computing counting statistics, as discussed in [36].



(a) Inverted Naïve Bayes (b) Triangle

Figure 4.1: Two models used to study network reduction. While inverted Naïve Bayes networks can have an arbitrary number of parents, we shall work with a three parent example.

3. The subgraph is an I-map over the remaining nodes

The first criteria ensures that ML is possible on the reduced network. The second criteria ensures that the query itself is not altered. For a query $Pr(\mathbf{Q}|\mathbf{E})$ if nodes in $\mathbf{Q}$ or $\mathbf{E}$ are disconnected, arcs must be added to ensure that this requirement is met. The third criteria is true when the original network is an I-map. However, these desiderata do not define an algorithm. Neither is there any theoretical result to argue that estimates produced using the reduced network will outperform MP. The goal of our experiments with network reduction is to determine whether it is a viable solution to learning with small data sets.

One example cited in support of this procedure is inverted Naïve Bayes (figure 4.1(a)) when the query of interest is $Pr(C)$. The number of possible configurations of the parents is exponentially large. Most configurations will not be observed in a small data set. Removing a few parents can reduce the number of configurations drastically. With the same training set, it eventually becomes possible to estimate all the parameters in the network. Ideally, this will lead to a better estimate of the query response.

We also introduce an even simpler problem, the triangle network (figure 4.1(b)). If the query of interest is $Pr(A|B)$ or $Pr(B|A)$ network reduction is simplified to the following choice – should node X be removed ?

4.5.3 Finite Mixture Models

There are two common criticisms of Laplacian smoothing. One, no consideration is given as to whether too much or too little probability mass is withheld for unseen events. Two, Laplacian smoothing is only a mixture of the maximum likelihood estimate and the uniform distribution. It may be advantageous to smooth using more elaborate mixture models. Herein we propose the use of *finite mixture models* (FMMs) to address both concerns.

Finite mixture models[4] represents a distribution as a weighted mixture of distributions from a chosen parametric family. Since CP-table rows in Bayesian networks are independent, each $\Theta_{v,x|\mathbf{f}} = Pr(X_v = x|f_1 \ldots f_k)$ can be treated as a single estimation problem. Our choice of classes is naïve: remove the last l conditioning events to create the l^{th} class. Note that we first impose a arbitrary ordering on the parents of each node. For notational simplicity we let f_0 act as a nonexistant conditioning event. Each class is estimated using either maximum likelihood or mean posterior. To calculate mixing parameters $\{\lambda_i\}_{i=0}^{k}$ expectation-maximization [16] on a hold-out set $D^h \subset D$ is used. Note that d^i denotes the value of variable X_i in sample d:

$$\textbf{E-Step:} \quad \beta_i = \sum_{d \in D^h} \frac{\lambda_i Pr(X_i = d^i | f_0 \ldots f_i)}{\sum_{i=0}^{k} \lambda_i Pr(X_i = d^i | f_0 \ldots f_i)} \tag{4.2}$$

$$\textbf{M-Step:} \quad \lambda_i = \frac{\beta_i}{\sum_{i=0}^{k} \beta_i} \tag{4.3}$$

[4]Commonly referred to as linear interpolation in natural language processing [34].

and the estimated local conditional probability distribution is as follows:

$$\hat{Pr}(X_v|f_1,\ldots,f_k) = \sum_{i=0}^{k} \lambda_i \cdot \hat{Pr}(X_v|f_1,\ldots,f_k) \tag{4.4}$$

One minor complication is how to deal with classes that have no data samples associated with them. While these classes can be simply removed, instead we use Laplacian smoothing when required. This prevents the algorithm from representing $\Theta_{v|\mathbf{f}}$ with only one or two classes simply because of a lack of data. Furthermore, finite mixture models are only used when the number of samples corresponding to $\Theta_{v|\mathbf{f}}$ is below a certain threshold.

This technique has been used in Hidden Markov Models for information extraction in text corpora [20] and interpolated Markov models for prokaryotic motif finding [42].

4.6 Experiments

4.6.1 Experiment Design

In this section we describe the design of two experiments. The first explores the viability of network reduction. The second evaluates the relative performance of algorithms for small sample estimation of the query response.

It has been claimed that inverted Naïve Bayes is a topology where network reduction will outperform MP – at least when the query of interest is $q(\Theta) = Pr(C)$. Network reduction under the constraints described in section 4.5.2 is straightforward: only remove parents of C. We shall assume all nodes are binary. The experiment begins with a noiseless sample of size 10. Network reduction removes parents of C, at random, until all the reduced network parameters can be specified by maximum likelihood. The value of $Pr(C = 1)$ on the reduced network is then returned. An MP estimate of the network parameters is produced using the original network. Finally, counting statistics can be used to produce an estimate solely from the data set. We average the results for each algorithm over 500 different data sets. If the expected L_2 error of network reduction is lower than the alternatives, we have reason to believe it will be a viable algorithm.

The second experiment uses a more realistic Bayesian network topology, Alarm [29], to compare our proposed algorithms. Our baseline is MP. The alternatives are finite mixture models on selected rows (FMMs), bagged MP estimators, and m-bagged MP estimators. In all scenarios the prior distribution over Θ is uniform.

The data flow of the second experiment is illustrated in figure 4.2. We are given the true distribution in the form of a Bayesian network. By assumption the graphical component is a minimal I-map. A single query is given. The true value of the query response, $q(\Theta^*)$, is derived by inference on the true distribution. This is the quantity each algorithm is trying to estimate. First, a noiseless data set of size n is generated. Next, this data set is used to produce t bootstrapped data sets for bagging and m-bagging. For bagging each data set is size n. For m-bagging each data set is size $m = \lfloor\sqrt{n}\rfloor$. Each algorithm produces an estimate of the query response. To mitigate (and measure) the effect of training sample variability the experiment is repeated over 75 independent data sets. The choice of 75 data sets is motivated by pragmatism rather than sampling theory. The convergence rate of EM in the FMM algorithm is quite slow, and several applications are required to learn a network. Moreover, we consider the behavior of these algorithms with 100 different queries.

4.6.2 Results

The results of network reduction on inverted Naïve Bayes (figure 4.3) show that, contrary to the claims made in [25], network reduction does not beat MP. In the first parameterization, network reduction often removes all the parents of node C. This is equivalent to ignoring the original structure altogether. Hence the similarity in expected L_2 error between counting statistics and network reduction.

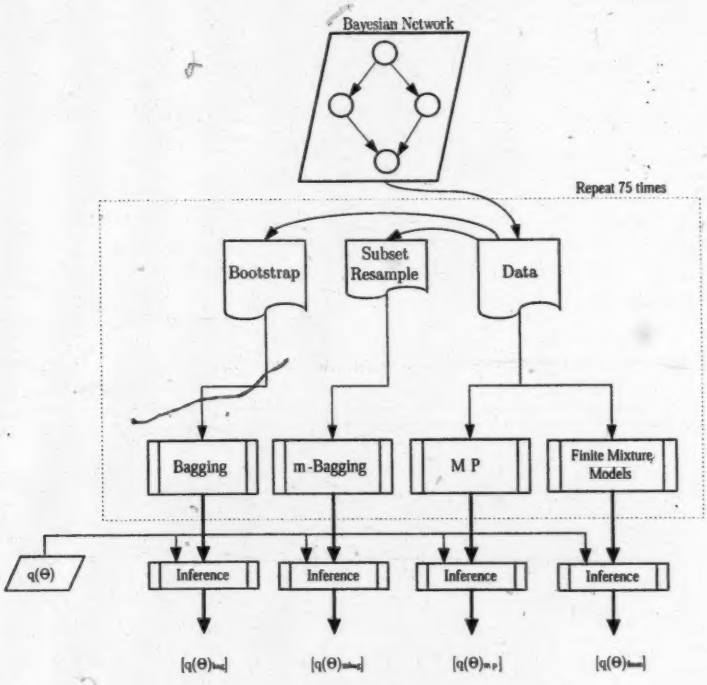
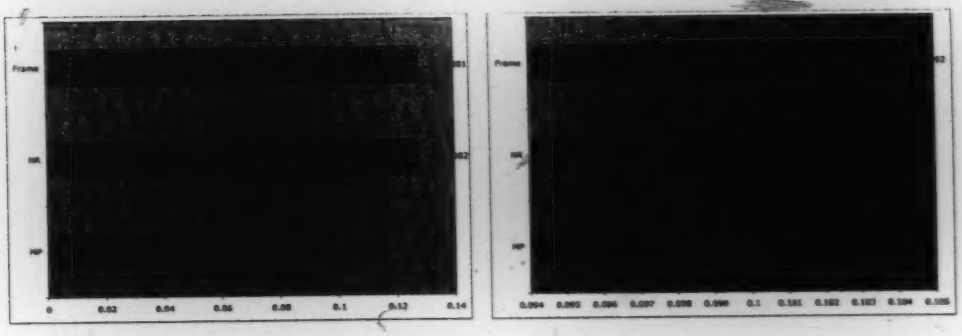


Figure 4.2: Experimental flow

To illustrate another limitation of network reduction, consider the triangle network in figure 4.1(b). We chose a single set of parameters Θ for the network and considered two queries: $Pr(A = true|B = false)$ and $Pr(B = false|A = true)$. In each case the only variable that can be removed is X. In every other respect the experimental setup is the same as for the inverted Naïve Bayes experiment. Network reduction beats MP on the first query; but loses to MP on the second query[5]. The algorithm could not decide whether or not to remove a single variable.

The results of the second experiment are presented in figures 4.4 and 4.5. Immediately clear is that certain queries have a higher sample complexity than others. This holds even when we consider two queries with similar expected responses – one often has a much higher squared error than the other. If we consider only the average squared error the following results arise. Bagging beats MP on 93 queries. m-bagging beats MP on 18 queries. FMM beats MP on 26 queries. Almost identical results hold when the sample size is increased to $n = 400$. However, it is our contention that one must also consider the stability of each algorithm to training set variation. FMMs can create overly sensitive estimators. Small perturbations in the mixing parameters can cause drastic changes in the conditional probability distributions. Given the tendency of EM to fall into local minima, the results are unsurprising. The use of EM introduces algorithmic variation, which is reflected in the stability of the estimator. In contrast, the aggregation estimators tend to provide increased stability. This holds moreso for bagging than for m-bagging as the latter is working with much smaller data sets. m-bagging tends to shrink the estimator towards the query response on the prior network, 0.5. A minority of the queries examined have responses near 0.5, which explains the poor performance of m-bagging against MP.

[5]The actual squared errors are as follows. For the first query network reduction had an error of 0.0049 compared to 0.0103 for MP. For the second query network reduction had an error of 0.0202 compared to 0.0129 for MP.



(a) Parameterization 1

(b) Parameterization 2

Figure 4.3: The performance of MP, network reduction, and counting statistics on two instances of the inverted Naïve Bayes topology with four parents. Data sets are size 10. `Frame` denotes the counting statistics algorithm, `NR` denotes the network reduction algorithm, and `MP` denotes the mean posterior algorithm. For parameterization 1 the true value of the query response is 0.568. For parameterization 2 the true value of the query response is 0.209. Reported results are averages over 500 data sets.

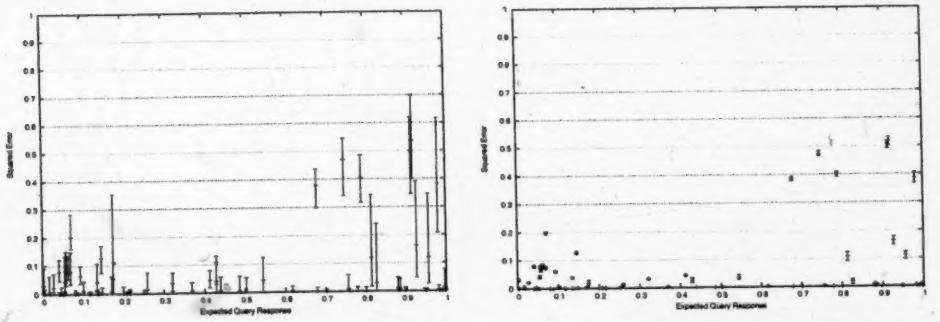
4.7 Discussion

The behavior of network reduction is vexing. It has been argued that node removal is analogous to marginalization of variables from a distribution function [25]. Since marginalization does not cause the query response to change, network reduction should work. Such reasoning is specious, using the mathematics of inference to justify reparameterization. Removing a node from the network indirectly changes the parameters involved in $q(\Theta)$. The problem is that the criteria used do not even attempt to minimize the error of the resulting estimator. Neither do these rules attempt to indirectly minimize the error between $q(\Theta)$ and $q(\Theta')$ by ensuring that the generative distributions are relatively close in a metric space. The criteria haphazardly attempt to approximate one function with another. Sometimes chance will be kind, and a better estimate is wrought. The same can be said when rolling dice.

If we knew when a data set will mislead MP, it might be possible to use network reduction as a fallback estimate. Consider the results for Alarm queries. Certain queries are accurately solved with a small number of samples; others are not. Cross-validation is commonly used to estimate the mean squared error of $q(\Theta)$. On small data sets this can lead to pessimistic estimates. Also, the estimate can itself have high variance [37]. Most importantly, cross-validation does not indicate whether the L_2 error is unusually large for data sets of that size. Neither can we use the variance of $q(\Theta)$, described in chapter 3. For small sample sizes the estimated query response $\hat{Pr}(\mathbf{Q}, \mathbf{E})/\hat{Pr}(\mathbf{E})$ is not an unbiased estimator of $Pr(\mathbf{Q}, \mathbf{E})/Pr(\mathbf{E})$. Finite sample bias is large enough that variance cannot be taken as an approximate measure of error. Finally, the marginal likelihood $Pr(D|\langle \mathcal{V}, \mathcal{A} \rangle)$ was applied in the hope that data which is improbable given the structure will produce poor estimates of the query response. Under the assumptions of this chapter this marginal can be computed using the *Bayesian scoring metric* [9]:

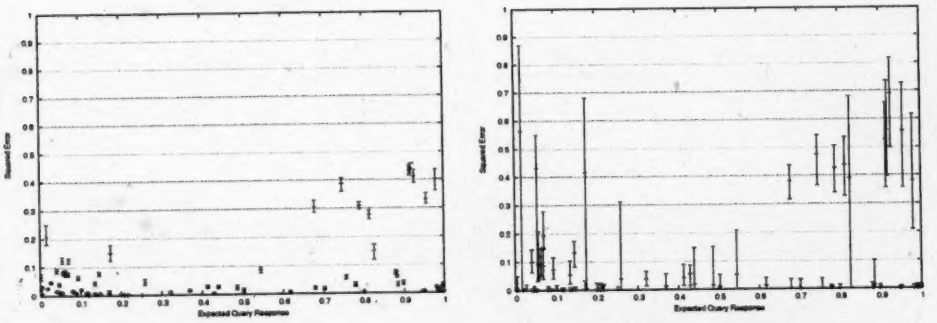
$$Pr(D|\langle \mathcal{V}, \mathcal{A} \rangle) = \prod_{v \in \mathcal{V}} \prod_{f \in F_v} \frac{\Gamma(\alpha_{v|\mathbf{f}})}{\Gamma(\alpha_{v|\mathbf{f}} + N_{v|\mathbf{f}})} \prod_{x \in X_v} \frac{\Gamma(\alpha_{v,x|\mathbf{f}} + N_{v,x|\mathbf{f}})}{\Gamma(\alpha_{v,x|\mathbf{f}})}$$

To prevent numerical overflow the log score is computed. Two queries with typical behavior are presented in figure 4.6. There is no significant correlation between the performance of MP and the data score. This result has also been observed on the Diamond network. The



(a) Mean Posterior

(b) Bagging



(c) m-Bagging

(d) Finite Mixture Model

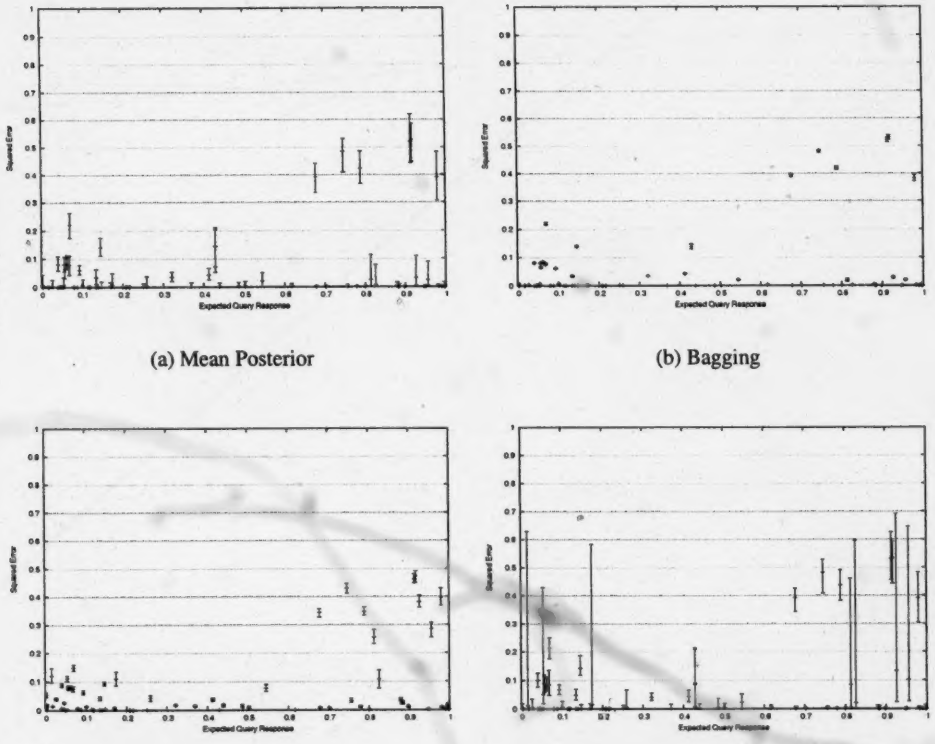
Figure 4.4: Squared error of queries given samples of size 100. Each point on the horizontal represents a query. Points represent the mean squared error (an average across 75 data samples). Vertical bars indicate the range of squared errors.

critical flaw is that while variables can have different degrees of influence on a query, all contribute to the data score with roughly equal effect. Even variables that are d-separated from the query nodes contribute to the data score.

The failure of finite mixture models as an alternative to MP is readily explained. The choice of classes is less than ideal. Randomly removing conditioning events produces simpler classes, but accurately estimating the mixing coefficients with small data sets is difficult. The EM step is optimizing $Pr(D|\lambda)$. Since the data set is small, the empirical distribution represented by D could be quite different from the true distribution. The introduction of stochasticity often increases the instability of the estimator to training set variation. Finally, the EM step makes this algorithm unbearably slow. Over 50 hours of CPU time was spent running these experiments, with the vast majority of the effort being allocated to solving EM problems.

Most promising are the bagging algorithms. It is not surprising that bagging tends to produce more accurate estimators than MP. Given the size of the data sets $q(\Theta)$ will have significant variance. Bagging is known to reduce variance. Another consequence of this tendency, previously unnoted in the literature, is that bagging is also stable in the face of training set variability. The L_2 error on one small data set is representative of bagging's performance on any equally sized data set.

The role of training set variability has generally been ignored by the machine learning community. This is understandable given the focus of many researchers on creating accurate



(a) Mean Posterior

(b) Bagging

(c) m-Bagging

(d) Finite Mixture Model

Figure 4.5: Squared error of queries given samples of size 400. Each point on the horizontal represents a query. Points represent the mean squared error (an average across 75 data samples). Vertical bars indicate the range of squared errors.

estimators rather than accurate algorithms. In the former scenario the goal is to produce an estimator that minimizes a loss function. In the latter scenario the goal is to determine whether one algorithm tends to produce better estimators than another. Accurate algorithms produce accurate estimators on a wide variety of problems and inputs – hence our focus on error across many equally sized data sets. Even if one focuses on a single problem one data set may make an algorithm look better than another; while other data sets lead to the reverse conclusion.

Our notion of accurate algorithms is prone to epistemic reduction. Averaging squared error over many data sets does not provide a rigorous statistical test of significance. Furthermore, current research [17, 37] indicates that proposed tests of significance either suffer from high type I error[6] or make unreasonable independence assumptions. Insofar as we can measure the superiority of learning algorithms, bagging MP estimators is advisable.

[6]That is, the null hypothesis is incorrectly rejected, and significance is falsely ascribed to the results.

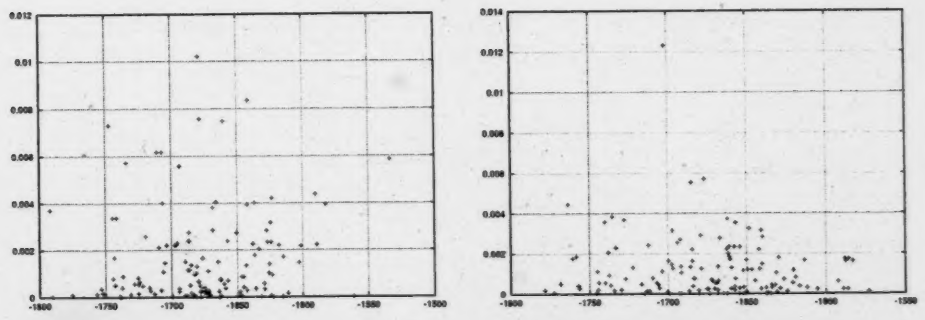


Figure 4.6: Data scoring for two queries on Alarm. The horizontal axis is $\log Pr(D|\langle \mathcal{V}, \mathcal{A} \rangle)$. The vertical axis is the L_2 error of an MP estimator of the query response. Each point represents this experiment with a different data set of size 100.

Chapter 5

Conclusion

The necessity of better procedures for estimation with small data sets is readily apparent. While the most elegant results of classical statistics often deal with large sample properties of estimators, the practice of machine learning often dictates the need for better techniques for small sample estimation in nonparametric models.

To that end we have studied discrete Bayesian networks where parameter uncertainty is encoded in Dirichlet row distributions. A common parameter estimation algorithm used in this scenario, mean posterior, produces a representation of parameter uncertainty which induces a distribution over the query response. In this paper we presented the question of whether the limitations of [45, 47] were due to the linear variance approximation or the normal approximation. Our results clearly show that for parameterizations learned with small data sets, the major problem is with the normal approximation. Furthermore, we have established the Beta distribution as a viable alternative model.

Continuing with the theme we have explored the problem of learning with small data sets in order to optimize a single query. We consider the problem in its most uncompromising formulation. A structure is given that may not have sufficient data to support accurate estimation and reparametrization of Θ is precluded. It has been suggested that altering the structure may lead to better estimates of the expected query response. A particular formulation of this idea was introduced in [25], but until now has not been experimentally tested. With simple examples we have illustrated the limitations of this technique. By projecting a function of all the network variables onto a lower dimensional space improvements may be had, but not with the criteria provided. Furthermore, we have explored other approaches and their viability when compared to the mean posterior technique. Finite mixture models represent an attempt to use hold-out smoothing as an alternative to Lidstonian smoothers. While it did not improve upon the results of MP, it illustrates the possibility of using data for smoothing in parameter learning. We further posed aggregation methods, of which bagging proved particularly promising. Not only does it almost always improve upon MP, but it is also resilient to training set variation. This behavior is, to our knowledge, unreported in the literature. This alludes to the epistemic question, "How do we know one algorithm is superior to another" ? We submit that these studies, and the questions they raise, constitute a meaningful contribution to the study of Bayesian networks.

Bibliography

[1] Yoshihisa Akimoto. A note on uniform asymptotic normality of the Dirchlet distribution. *Mathematica Japonica*, 44(1):25–30, Dec 1996.

[2] Kai Oliver Arras. An introduction to error propagation: Derivation, meaning, and examples of equation $C_y = F_x C_x F_x^T$. Technical Report EPFL-ASL-TR-98-01 R3, Autonomous Systems Lab, Institute of Robotic Systems, Swiss Federal Institute of Technology Lausanne, Sept 1998.

[3] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, Boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.

[4] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford U.P., 1998.

[5] Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in Bayesian networks. In *Proceedings of the Twelvth Annual Conference on Uncertainty in Artificial Intelligdence (UAI-96)*, pages 115–123, 1996.

[6] Leo Breiman. Bagging predictors. Technical Report No. 421, Dept. of Statistics, University of California, Sep 1994.

[7] Hei Chan and Adnan Darwiche. When to numbers really matter? *Journal of Artificial Intelligence Research*, 17, 2002.

[8] Jian Cheng and Marek J. Druzdzel. Confidence inference in Bayesian networks. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-2001)*, pages 75–82. Morgan Kaufmann Publishers, 2001.

[9] Greg Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning Journal*, 9:309–347, 192.

[10] Gregory F. Cooper. Probabilistic inference using belief networks is NP-Hard. Technical Report KSL-87-27, Knowledge Systems Laboratory, Stanford University, 1987.

[11] P. Dagum and M. Luby. Approximating probabilistic inference in bayesian belief networks is NP-hard. *Artificial Intelligence*, 60:141–153, 1993.

[12] D.A. Darling. The Kolmogorov-Smirnov, Cramer-von Mises tests. *Ann. Math. Stat*, 28:823–838, 1957.

[13] Adnan Darwiche. A differential approach to inference in Bayesian networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, 2000.

[14] Adnan Darwiche and Gregory M. Provan. Query DAGs: A practical paradigm for implementing belief network inference. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, 1996.

[15] Rina Dechter. Bucket elimination: A unifying framework for probabilistic inference. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pages 211–219, 1996.

[16] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Series B*, 39:1–38, 1977.

[17] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[18] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

[19] Enrico Fagiuoli and Marco Zaffalon. 2U: An exact interval propagation algorithm for polytrees with binary variables. *Artificial Intelligence*, 106:77–107, 1998.

[20] Dayne Freitag and Andrew Kachites McCallum. Information extraction with HMMs and shrinkage. In *Proc. AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.

[21] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

[22] Dan Geiger and David Heckerman. Learning Gaussian networks. Technical Report MSR-TR-94-10, Microsoft Research, 1994.

[23] Subhashis Ghosal. Convergence rates for density estimation with Bernstein polynomials. *Annals of Statistics*, 29(5), Oct 2001.

[24] Russ Greiner, Adam Grove, and Dale Schuurmans. Learning Bayesian nets that perform well. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, 1997.

[25] Russell Greiner, Dale Schuurmans, and Chris O'Brien. Efficient estimation exploiting independence constraints. Unpublished Manuscript.

[26] Russell Greiner and Wei Zhou. Structural extension to logistic regression: Discriminant parameter learning of belief net classifiers. In *Proceedings of the Eighteenth Annual National Conference on Artificial Intelligence (AAAI-02)*, pages 167–173, Aug 2002.

[27] David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Technical Report MSR-TR-94-09, Microsoft Research, 1994.

[28] David E. Heckerman. A tutorial on learning with Bayesian networks. In M. I. Jordan, editor, *Learning in Graphical Models*, 1998.

[29] Edward Herskovits and Gregory Cooper. Algorithms for Bayesian belief network precomputation. *Methods of Information in Medicine*, pages 362–370, 1991.

[30] J. Kim and J. Pearl. A computational model for causal and diagnostic reasoning in inference engines. In *Proceedings Eigth International Joint Conference on Artificial Intelligence (IJCAI-83)*, pages 190–193, 1983.

[31] Gernot D. Kleiter. Propagating imprecise probabilities in Bayesian networks. *Artificial Intelligence*, 88:143–161, 1996.

[32] Steffen L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201, 1995.

[33] David D. Lewis. Naïve (Bayes) at forty: The independence assumption in information retrieval. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, number 1398, pages 4–15, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[34] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2001.

[35] Andrew W. Moore. Private communication, May 2003.

[36] Andrew W. Moore and Mary Soon Lee. Cached sufficient statistics for efficient machine learning with large datasets. *Journal of Artificial Intelligence Research*, 8:67–91, 1997.

[37] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine Learning*, 52(3):239–281, Sep 2003.

[38] Radford M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113, 1992.

[39] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes. In *NIPS-02*.

[40] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman Publishers, San Mateo, CA, 1988.

[41] Eric Sven Ristad. A natural law of succession. Technical Report CS-TR-495-95, Princeton University, 1995.

[42] S. Salzberg, A. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated markov models. *Nucleic Acids Research*, 26(2):544–548, 1998.

[43] David J. Spiegelhalter. A unified approach to imprecision and sensitivity of beliefs in expert systems. In *Proceedings of the Third Conference on Uncertainty in Artificial Intelligence*, pages 199–208, 1989.

[44] David J. Spiegelhalter and Steffen L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, pages 579–605, 1990.

[45] Tim van Allen. Handling uncertainty when you're handling uncertainty: Model selection and error bars for belief networks. Master's thesis, Dept. of Computing Science, University of Alberta, 2000.

[46] Tim van Allen and Russell Greiner. Error-bars for belief net inference. Unpublished Manuscript, Mar 2001.

[47] Tim van Allen, Russell Greiner, and Peter Hooper. Bayesian error-bars for belief net inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, Aug 2001.

[48] Robert L. Winkler and William L. Hays. *Statistics: Probability, Inference, and Decision*. Holt, Rinehart, and Winston, 2nd edition, 1975.

Appendix A

Multivariate Delta Rule

Herein we present a derivation of the approximate mean and variance for any differentiable function of random variables, $f(X_1, \ldots, X_n)$. The presentation is based largely on [2] and is included here for completeness. Consider an approximation to $f(X_1, \ldots, X_n)$, namely the first-order Taylor expansion around $(\mu_1, \ldots, \mu_n)$.

$$Y = f(\mu_1, \ldots, \mu_n) + \sum_{i=1}^{n} \left[\frac{\partial f}{\partial X_i}(\mu_1, \ldots, \mu_n) \right] [X_i - \mu_i]$$

Let

$$\begin{aligned} a_0 &= f(\mu_1, \ldots, \mu_n) \\ a_i &= \frac{\partial f}{\partial X_i}(\mu_1, \ldots, \mu_n) \end{aligned}$$

The mean of variance of Y are approximations to the respective moments of $f(X_1, \ldots, X_n)$

$$\begin{aligned} \mu_Y \doteq E[Y] &= E[a_0 + \sum_i a_i(X_i - \mu_i)] \\ &= E[a_0] + E[\sum_i a_i(X_i - \mu_i)] \\ &= a_0 + \sum_i (a_i E[X] - a_i E[\mu_i]) \\ &= a_0 + \sum_i (a_i \mu_i - a_i \mu_i) \\ &= f(\mu_1, \ldots, \mu_n) \end{aligned}$$

$$\begin{aligned} \sigma_Y^2 \doteq E[(Y - E[Y])^2] &= E[(\sum_i a_i(X_i - \mu_i))^2] \\ &= E[\sum_i a_i^2 (X_i - \mu_i)^2 + \sum_{i \neq j} \sum a_i a_j (X_i - \mu_i)(X_j - \mu_j)] \\ &= \sum_i a_i^2 E[(X_i - \mu_i)^2] + \sum_{i \neq j} \sum a_i a_j E[(X_i - \mu_i)(X_j - \mu_j)] \\ &= \sum_i a_i^2 \sigma_i^2 + \sum_{i \neq j} \sum a_i a_j \sigma_{ij} \\ &= f'(\mu) \cdot \Sigma \cdot f'(\mu)^T \end{aligned}$$

where

$$f'(\mu) \doteq \left[\frac{\partial f}{\partial X_1}(\mu), \ldots, \frac{\partial f}{\partial X_n}(\mu)\right]$$

and Σ is the variance-covariance matrix of $\{X_1, \ldots, X_n\}$.

Appendix B

Description of InfEB

In this section we provide pseudo-code for the InfEB algorithm for variance estimation depicted in figure 3.1. The descriptions herein depend heavily on notation introduced in section 3.4.

Algorithm 1: Initialize data structures for BE$^+$
Input: Variable ordering π, network CP-tables $\{h_1(S_1), \ldots, h_n(S_n)\}$, and a set of variable assignments $\mathbf{A} = \mathbf{a}$
Output: Operation stack $\mathcal{S}$ and bucket list $b_0 \ldots b_n$
Description: π_j denotes variable X_j in the variable ordering. π_0 denotes a variable not in the network
INITIALIZE(π, $\{h_1(S_1), \ldots, h_n(S_n)\}$, $\mathbf{A} = \mathbf{a}$)

(1) Create a list of bucket where b_i corresponds to variable X_i. b_0 is the nil bucket. All buckets are initially empty sets
(2) **for** $i = 1$ **to** n
(3) $\quad h'_i(S'_i) \leftarrow h_i(a)$ (*i.e.* Apply evidence)
(4) $\quad j \leftarrow \max\{j : 0 \leq j \leq n \wedge \pi_j \in S'_i\}$
(5) $\quad b_j \leftarrow b_j \cup \{h'_i(S'_i)\}$
(6) $\quad$ push record $(h_i(S_i), h'_i(S'_i), noBucket, j)$ onto $\mathcal{S}$
(7) **return** $b_0, \ldots, b_n$, $\mathcal{S}$

Note that the records pushed onto $\mathcal{S}$ in this algorithm indicate that the function $h'_i(S'_i)$ was not created by processing a bucket, but by applying evidence to $h_i(S_i)$ and placing it in bucket b_j.

Algorithm 2: Calculate expected marginal probability and partial derivatives of the marginal w.r.t. each network parameter
Input: Bucket list $b_0 \dots b_n$, Operation stack $\mathcal{S}$
Output: Marginal probability $Pr(\mathbf{A} = \mathbf{a})$, Derivative stack $\mathcal{D}$, Function stack $\mathcal{P}$
Description: While naïve implementations of BE$^+$might keep all partial derivatives in memory, only the most recent one w.r.t. a given CP-table needs to be stored.
BE$^+(b_0, \dots, b_n, \mathcal{S})$
(1) Let $\mathcal{D} \leftarrow \emptyset$ be a stack of functions.
(2) Let $\mathcal{P} \leftarrow \emptyset$ be a stack of Dirichlet CP-tables
(3) **for** $i = n$ **to** 1
(4) **if** $b_i \neq \emptyset$
(5) $f(T) \leftarrow \text{elim}_{X_i}[\text{join}(b_i)]$
(6) $j \leftarrow \max\{j : 0 \leq j < i \wedge \pi_j \in T\}$
(7) $b_j \leftarrow b_j \cup \{f(T)\}$
(8) push record $(nil, f(T), i, j)$ onto $\mathcal{S}$
(9) Initialize all elements of Deriv$[1 \dots n]$ to the unity function (n is the number of nodes in the network)
(10) **while** $\mathcal{S} \neq \emptyset$
(11) $(q(L), f(T), i, j) \leftarrow \text{pop}(\mathcal{S})$
(12) $g(R) \leftarrow \text{elim}_{L-T}[join(f(T) \cup b_j - \text{Deriv}[j])]$
(13) **if** $i = noBucket$
(14) push $g(R)$ onto $\mathcal{D}$
(15) push $q(L)$ onto $\mathcal{P}$
(16) **else**
(17) Deriv$[i] \leftarrow g(R)$
(18) $Pr(\mathbf{a}) \leftarrow \text{join}(b_0)$
(19) **return** $Pr(\mathbf{a}), \mathcal{D}, \mathcal{P}$

Algorithm 3: Estimate the variance of query $Pr(\mathbf{q}|\mathbf{e})$
Input: For each marginal probability $p = Pr(\mathbf{q}, \mathbf{e})$ and $q = \Pr(\mathbf{e})$, the outputs of BE$^+$. Also, the evidence associated with each marginal: $\mathbf{Q} = \mathbf{q}, \mathbf{E} = \mathbf{e}$.
Output: Estimated variance of the query response: $\hat{\sigma}^2_{Pr(\mathbf{q}|\mathbf{e})}$
Description: Note that conforms[$\cdot, \cdot$] returns true iff the two arguments do not disagree on any variable assignment

ESTIMATEVARIANCE(...)

(1) $variance \leftarrow 0$
(2) **while** $\mathcal{P}_{Pr(\mathbf{Q},\mathbf{E})}$ not empty
(3) $f(A) \leftarrow$ pop $\mathcal{P}_{Pr(\mathbf{Q},\mathbf{E})}$
(4) $g(B) \leftarrow$ pop $\mathcal{P}_{Pr(\mathbf{E})}$
(5) $h(C) \leftarrow$ pop $\mathcal{D}_{Pr(\mathbf{E})}$
(6) Let X and $\mathbf{Y}$ be the dependent and conditioning variables of $h(C)$
(7) **foreach** assignment to $\mathbf{Y}, h(\mathbf{y})$
(8) Calculate covariance matrix Σ for current row
(9) **foreach** values of X (denoted x)
(10) **if** conforms[$\mathbf{y} : x, q : e$]
(11) $dp \leftarrow f(\mathbf{y} : x)$
(12) **else**
(13) $dp \leftarrow 0$
(14) **if** confirms[$\mathbf{y} : x, e$]
(15) $dq \leftarrow g(\mathbf{y} : x)$
(16) **else**
(17) $dq \leftarrow 0$
(18) $PD[i] \leftarrow (q \cdot dp - p \cdot dq)/q^2$ (*i.e.* $\frac{\partial q(\Theta)}{\partial \Theta_{v,x|\mathbf{f}}}$)
(19) $variance \leftarrow variance + PD \cdot \Sigma \cdot PD^T$
(20) **return** $variance$

Appendix C

Variance of $q(\Theta)$ under Naïve Bayes

In this section we derive a specialized form of equation 3.4 for Naïve Bayes networks that precludes the need for partial derivatives of the query. Consider the covariance matrix of $\Theta_{v|\mathbf{f}} \sim Dirichlet(\alpha_{v,1|\mathbf{f}}, \ldots, \alpha_{v,r|\mathbf{f}})$. For notational simplicity we let

$$\begin{aligned} \alpha_{v|\mathbf{f}} &= \sum_{x=1}^{r} \alpha_{v,x|\mathbf{f}} \\ \Theta^*_{v,x|\mathbf{f}} &= \frac{\alpha_{v,x|\mathbf{f}}}{\alpha_{v|\mathbf{f}}} \\ \Theta^*_{v|\mathbf{f}} &= (\Theta^*_{v,1|\mathbf{f}}, \ldots, \Theta^*_{v,r|\mathbf{f}}) \end{aligned}$$

Given a Naïve Bayes network with root node H, evidenced children $\mathbf{E} = \{E_1, \ldots, E_k\}$, and unevidenced children $\{X_1, \ldots, X_t\}$ we seek to calculate the variance of query response $q(\Theta) = Pr(H = h|\mathbf{E} = \mathbf{e}) = Pr(h|\mathbf{e})$. Equation 3.6 is impractically slow in general. However, for Naïve Bayes networks we derive a tractable form. We consider the variance contribution of the class node, a row in an unevidenced child node, and a row in an evidenced child node independently. For notational compactness we let

$$C(h) = \left[Pr(h|\mathbf{e}) - Pr(h|\mathbf{e})^2\right]^2$$

noting that when H is binary $C(True) = C(False)$. This allows even further simplification of the following equations.

Class node: There are no parents for this node, so $\mathbf{f} = \emptyset$. By substitution into equation 3.6 and straightforward algebra the contribution of the class node is as follows:

$$\sigma^2_{Pr(H)} = \frac{1}{m_{H|\emptyset} + 1} \left[\sum_h \frac{C(h)}{\Theta^*_{h|\emptyset}}\right]$$

Row in an evidenced child: There is only one parent, $\mathbf{f} = H$. We take advantage of the fact that for an evidence node $E_j = e_j$

$$Pr(E_j, h|\mathbf{e}) = \begin{cases} Pr(h|\mathbf{e}), & \text{if } E_j = e_j \\ 0, & \text{otherwise} \end{cases}$$

By substitution into equation 3.6

$$\begin{aligned}
\sigma^2_{Pr(E_j|h)} &= \frac{1}{m_{E_j|h}+1}\left[\sum_i \frac{1}{\Theta^*_{e_i|h}}\left[Pr(e_i,h|\mathbf{e}) - Pr(h|\mathbf{e})Pr(e_i,h|\mathbf{e})\right]^2 - C(h)\right] \\
&= \frac{1}{m_{E_j|h}+1}\left[\frac{1}{\Theta^*_{e_j|h}}C(h) - C(h)\right] \\
&= \frac{1}{m_{E_j|h}+1}\left[C(h)\left(\frac{1}{Pr(e_j|h)} - 1\right)\right]
\end{aligned}$$

Row in a nonevidenced child: There is only one parent, $\mathbf{f} = H$. We take advantage of the generalized product rule:

$$\begin{aligned}
Pr(x_i,h|\mathbf{e}) &= Pr(x_i|h,\mathbf{e})Pr(h|\mathbf{e}) \\
&= Pr(x_i|h)Pr(h|\mathbf{e})
\end{aligned}$$

By substitution into equation 3.6

$$\begin{aligned}
\sigma^2_{Pr(X_j|h)} &= \frac{1}{m_{X_j|h}+1}\left[\sum_i \frac{1}{\Theta^*_{x_i|h}}\left[Pr(x_i,h|\mathbf{e}) - Pr(h|\mathbf{e})Pr(x_i,h|\mathbf{e})\right]^2 - C(h)\right] \\
&= \frac{1}{m_{X_j|h}+1}\left[\sum_i \frac{1}{\Theta^*_{x_i|h}}\left[Pr(x_i|h)Pr(h|\mathbf{e}) - Pr(h|\mathbf{e})Pr(x_i|h)Pr(h|\mathbf{e})\right]^2 - C(h)\right] \\
&= \frac{1}{m_{X_j|h}+1}\left[\sum_i \frac{Pr(x_i|h)^2}{\Theta^*_{x_i|h}}C(h) - C(h)\right] \\
&= \frac{1}{m_{X_j|h}+1}\left[\overbrace{\sum_i Pr(x_i|h)}^{1} C(h) - C(h)\right] \\
&= 0
\end{aligned}$$

To calculate the overall variance of the query response simply sum the variance contribution from the query node with the variance contribution from each evidenced child node. The only inputs required are the network parameters, the effective samples size of each CP-table row, and the expected query response. There are no additional space requirements. The algorithmic time complexity is linear in the size of the network.

Appendix D

Fisher Information on Bayesian Networks

Given an I-map of the true distribution and iid data $D = \{d_1 \dots d_N\}$ we want to find the maximum likelihood estimator of network parameters Θ. Let $\Theta_{v,x|\mathbf{f}} = Pr\{X_i = x|\mathbf{F}_v = \mathbf{f}\}$ and $N_{v,x|\mathbf{f}}$ be the corresponding counts in the data. The maximum likelihood estimator of Θ is

$$\begin{aligned}
\hat{\Theta} &= \underset{\Theta}{\text{argmax}}\,[P(D|\Theta)] \\
&= \underset{\Theta}{\text{argmax}}\,[\log P(D|\Theta)] \\
&= \underset{\Theta}{\text{argmax}}\left[\log \prod_{v|\mathbf{f}} \prod_{x} \Theta_{v,x|\mathbf{f}}^{N_{v,x|\mathbf{f}}}\right] \\
&= \underset{\Theta}{\text{argmax}}\left[\sum_{v|\mathbf{f}} \sum_{x} N_{v,x|\mathbf{f}} \log \Theta_{v,x|\mathbf{f}}\right]
\end{aligned}$$

For notational simplicity we denote the log-likelihood $l = \log P(D|\Theta)$ and Θ_u and Θ_v to be two network parameters. The Fisher information $F(\Theta)$ is as follows:

$$\begin{aligned}
\frac{\partial l}{\partial \Theta_u} &= \sum_{v|\mathbf{f}} \sum_{x} \frac{N_{v,x|\mathbf{f}}}{\Theta_{v,x|\mathbf{f}}} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_u} \\
F(\Theta) = E\left[-\frac{\partial^2 l}{\partial \Theta_u \partial \Theta_v}\right] &= -\sum_{v|\mathbf{f}} \sum_{x} \frac{N_{v,x|\mathbf{f}}}{\Theta_{v,x|\mathbf{f}}} \cdot \frac{\partial^2 \Theta_{v,x|\mathbf{f}}}{\partial \Theta_u \partial \Theta_v} + \sum_{v|\mathbf{f}} \sum_{x} \frac{N_{v,x|\mathbf{f}}}{\Theta_{v,x|\mathbf{f}}^2} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_u} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_v} \\
&= -\sum_{v|\mathbf{f}} \sum_{x} \frac{N\Theta_{v,x|\mathbf{f}}}{\Theta_{v,x|\mathbf{f}}} \cdot \frac{\partial^2 \Theta_{v,x|\mathbf{f}}}{\partial \Theta_u \partial \Theta_v} + \sum_{v|\mathbf{f}} \sum_{x} \frac{N\Theta_{v,x|\mathbf{f}}}{\Theta_{v,x|\mathbf{f}}^2} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_u} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_v} \\
&= -N \sum_{v|\mathbf{f}} \sum_{x} \frac{\partial^2 \Theta_{v,x|\mathbf{f}}}{\partial \Theta_u \partial \Theta_v} + N \sum_{v|\mathbf{f}} \sum_{x} \frac{1}{\Theta_{v,x|\mathbf{f}}} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_u} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_v} \\
&= -N \sum_{v|\mathbf{f}} \frac{\partial^2 \overbrace{\Sigma_x \Theta_{v,x|\mathbf{f}}}^{1}}{\partial \Theta_u \partial \Theta_v} + N \sum_{v|\mathbf{f}} \sum_{x} \frac{1}{\Theta_{v,x|\mathbf{f}}} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_u} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_v}
\end{aligned}$$

$$= N \sum_{v|\mathbf{f}} \sum_{x} \frac{1}{\Theta_{v,x|\mathbf{f}}} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_u} \cdot \frac{\partial \Theta_{v,x|\mathbf{f}}}{\partial \Theta_v}$$

The Cramér-Rao variance lower bound for an unbiased estimator is just the inverse matrix of the Fisher information. Calculation of this quantity is possible. The derivatives required can be produced using BE^+. It should be noted that this is a variance estimator for the generative distribution; not the query response. However, when $q(\Theta)$ is injective the invariance property of maximum likelihood can be used to extend this result to estimate the variance of $q(\Theta)$.